

Pontificia Universidad Católica del Ecuador

Facultad De Ingeniería

Escuela de Sistemas



TEMA:

Desarrollar una aplicación web responsiva para la gestión de reservas y administración de restaurante.

AUTOR:

SEBASTIÁN ANDRÉS CADENA MOLINA

TRABAJO PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS DE INFORMACIÓN

QUITO DM, 26 DE JUNIO DE 2023

A la Sabiduría y la Serenidad de mi abuelo, ejemplo que la perseverancia vence todo obstáculo y adversidad.

Al Apoyo, la Fuerza y el Amor Incondicional de mi Madre, Tía y Hermano, los cimientos de mi mayor fortaleza y la fuente inagotable de inspiración en mi búsqueda de aspiraciones y metas.

HOMO PROPOSITUM ANIMUM INDIGET

AGRADECIMIENTO

La culminación del presente trabajo académico ha sido posible gracias al apoyo de personas, que me prestaron su tiempo, confiaron en mi persona y me brindaron su mano cuando lo necesite. Agradezco a la ayuda incondicional mi familia quienes supieron motivarme y levantarme día a día en la realización de la presente disertación. A todo el cuerpo docente de la Pontificia Universidad Católica del Ecuador por los conocimientos impartidos y colaboraron en la realización del presente trabajo.

Y un agradecimiento muy especial al Ingeniero Miguel Ortiz Navarrete, por sus valiosas enseñanzas: mi respeto, admiración, cariño y gratitud eternos.

RESUMEN

El presente proyecto tiene como objetivo el desarrollo de una aplicación web responsiva para la gestión de reservas y administración de restaurante. El capítulo I, titulado "Introducción", establece el marco de referencia de la investigación. En primer lugar, se justifica la necesidad de esta aplicación, destacando la importancia de optimizar los procesos de reserva y administración en el restaurante. A continuación, se plantea el problema a resolver, que se centra en la falta de una herramienta eficiente para gestionar y administrar las reservas en los establecimientos gastronómicos. El objetivo general del proyecto es desarrollar una aplicación web que solucione esta problemática, y se detallan los objetivos específicos que guiarán el proceso.

El capítulo I también presenta los antecedentes relacionados con el tema, donde se destacan investigaciones y desarrollos previos relacionados con la gestión de reservas y la administración del restaurante. Además, se establece el alcance del proyecto, definiendo las funcionalidades y características que se abordarán en la aplicación.

El capítulo II, denominado "Fundamentación Teórica", se centra en proporcionar los fundamentos conceptuales necesarios para comprender el desarrollo de la aplicación. Se abordan temas como las páginas web, HTML, CSS, JavaScript y PHP, que son tecnologías fundamentales para la implementación de la aplicación web. También se analiza la arquitectura de software y se presentan diferentes metodologías de desarrollo de software, incluyendo la metodología aplicada en este proyecto. Se introduce a la metodología SCRUM, se exploran las etapas y se comparan diferentes metodologías ágiles de desarrollo. Por último, se presentan las herramientas tecnológicas que se utilizarán en el desarrollo de la aplicación.

El capítulo III, titulado "Metodología", detalla la metodología de desarrollo de la tesis. Se explica la metodología aplicada, que se basa en el enfoque ágil SCRUM. Se presenta la

metodología para el desarrollo de software, que establece las etapas y los procesos que se seguirán en el proyecto. Se realiza un análisis comparativo de diferentes metodologías y se destacan las metodologías ágiles de desarrollo. Además, se presentan las herramientas tecnológicas que se utilizarán a lo largo del proyecto.

El capítulo IV, "Desarrollo de la Investigación", se enfoca en la aplicación práctica de la metodología SCRUM. Se identifican los roles necesarios para el desarrollo del proyecto y se detallan los requerimientos funcionales y no funcionales de la aplicación. Luego, se aborda el diseño lógico y funcional de la aplicación, definiendo la estructura y la interfaz de usuario.

El capítulo V, "Implementación", se centra en la implementación de la aplicación. Se detallan las etapas del proyecto y se describen los diferentes sprints que se realizarán para desarrollar gradualmente la aplicación.

Finalmente, en las conclusiones se resumen los principales hallazgos y resultados del proyecto. Se presentan las recomendaciones para mejorar la aplicación y se destacan las contribuciones del proyecto en el ámbito de la gestión de reservas y administración del restaurante.

ÍNDICE

| | |
|--|----|
| CAPÍTULO I: INTRODUCCIÓN..... | 1 |
| 1. MARCO DE REFERENCIA..... | 1 |
| 1.1. Justificación..... | 1 |
| 1.2. Planteamiento del problema..... | 2 |
| 1.3. Objetivo General..... | 2 |
| 1.4. Objetivos Específicos..... | 3 |
| 1.5. Antecedentes..... | 3 |
| 1.6. Alcance..... | 4 |
| CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA..... | 5 |
| 2. Marco Teórico..... | 5 |
| 2.1. Página Web..... | 5 |
| 2.2. HTML..... | 5 |
| 2.3. CCS..... | 5 |
| 2.4. Javascript..... | 6 |
| 2.5. PHP..... | 7 |
| 2.6. Arquitectura de Software..... | 8 |
| 2.6.1. Modelo Vista Controlador (MVC)..... | 9 |
| 2.7. Metodología de Aplicada..... | 9 |
| 2.8. Metodología Desarrollo de Software..... | 10 |

| | |
|--|-----------|
| 2.8.1. SCRUM..... | 10 |
| 2.9. UML..... | 11 |
| 2.10. Base de Datos | 12 |
| 2.10.1. Gestor de Base de Datos | 13 |
| 2.10.2. Base de Datos Relacional | 13 |
| 2.10.3. MYSQL | 14 |
| 2.11. IDE | 15 |
| 2.12. Diagrama..... | 15 |
| 2.12.1. Diagrama lógico | 16 |
| 2.12.2. Diagrama físico..... | 16 |
| 2.12.3. Diagrama conceptual | 17 |
| 2.12.4. Diagrama de clases | 17 |
| 2.12.5. Diagramas de casos de uso | 18 |
| 2.12.6. Diagrama de secuencia..... | 18 |
| CAPÍTULO III: METODOLOGÍA..... | 20 |
| 3. Metodología de desarrollo de la tesis..... | 20 |
| 3.1. Metodología Aplicada | 20 |
| 3.2. Metodología para el Desarrollo de Software | 21 |
| 3.2.1. Metodologías de Desarrollo Tradicionales..... | 22 |
| 3.2.2. Metodologías de Desarrollo Ágiles | 24 |
| 3.3. Análisis, Etapas y Comparativa de las Metodologías..... | 25 |

| | |
|---|----|
| 3.4. Metodologías Ágiles de Desarrollo | 28 |
| 3.4.1. SCRUM..... | 29 |
| 3.4.1.1. Roles | 30 |
| 3.4.1.2. Artefactos | 30 |
| 3.4.1.3. Evaluación de los Criterios Establecidos..... | 32 |
| 3.4.2. Extreme Programming (XP) | 32 |
| 3.4.2.1. Características..... | 32 |
| 3.4.2.2. Roles. | 34 |
| 3.4.2.3. Evaluación de los Criterios Establecidos..... | 35 |
| 3.4.3. Comparativa de las Metodologías Ágiles..... | 35 |
| 3.5. Herramientas Tecnológicas | 36 |
| 3.5.1. Entorno de desarrollo de código..... | 37 |
| 3.5.1.1. Visual Studio Code..... | 37 |
| 3.5.1.2. Comparación de entornos de Desarrollo Web..... | 37 |
| 3.5.2. Base de Datos..... | 39 |
| 3.5.2.1. MySQL..... | 39 |
| 3.5.2.2. Ventajas y Desventajas | 39 |
| 3.5.2.3. Conclusión..... | 40 |
| 3.5.3. Repositorio..... | 40 |
| 3.5.3.1. GitHub | 40 |
| 3.5.3.2. Ventajas y Desventajas | 41 |

| | |
|--|----|
| 3.5.3.3. Conclusión | 42 |
| CAPÍTULO IV: DESARROLLO DE LA INVESTIGACIÓN..... | 43 |
| 4. Aplicación de la Metodología SCRUM..... | 43 |
| 4.1. Identificación de Roles..... | 43 |
| 4.2. Identificación de Requerimientos | 44 |
| 4.2.1. Reunión Product Owner | 44 |
| 4.2.2. Requerimientos Funcionales | 45 |
| 4.2.2.1. Administrador del Restaurante..... | 46 |
| 4.2.2.2. Cliente del Restaurante | 47 |
| 4.2.3. Requerimientos no funcionales | 47 |
| 4.2.4. Product Backlog | 48 |
| 4.2.5. Sprint Planning Meeting | 49 |
| 4.3. Diseño Lógico y Funcional..... | 51 |
| 4.3.1. Diagrama General de Casos de Uso | 51 |
| 4.3.2. Nivel de Detalle de Casos de Uso | 52 |
| 4.3.2.1. RF1 Gestionar Plato | 52 |
| 4.3.2.2. RF2 Gestionar Pedido | 58 |
| 4.3.2.3. RF3 Gestionar Reservas..... | 66 |
| 4.3.2.4. RF4 Gestionar Producto | 74 |
| 4.3.2.5. RF6 Gestionar Empleado..... | 81 |
| 4.3.2.6. RF7 Gestionar Cliente | 88 |

| | | |
|----------------------------------|---|-----|
| 4.3.3. | Diagramas de Secuencia | 95 |
| 4.3.3.1. | RF1 Gestionar Plato | 95 |
| 4.3.3.2. | RF2 Gestionar Pedido | 99 |
| 4.3.3.3. | RF3 Gestionar Reservas..... | 105 |
| 4.3.3.4. | RF4 Gestionar Producto | 109 |
| 4.3.3.5. | RF6 Gestionar Empleado..... | 114 |
| 4.3.3.6. | RF7 Gestionar Cliente | 119 |
| 4.3.4. | Diagrama de Clases | 1 |
| 4.3.5. | Diagrama Conceptual de Base de Datos..... | 1 |
| 4.3.6. | Diagrama Lógico de Base de Datos | 2 |
| 4.3.7. | Diagrama Físico de Base de Datos | 3 |
| 4.3.8. | Arquitectura de La Interfaz | 1 |
| CAPÍTULO V: IMPLEMENTACIÓN | | 2 |
| 5. | Implementación de la aplicación | 2 |
| 5.1. | Implementación de la tesis | 2 |
| 5.1.1. | Productos Entregados..... | 2 |
| 5.1.1.1. | Diseño de la Base de Datos..... | 2 |
| 5.1.1.2. | Base de Datos | 2 |
| 5.2. | Sprint..... | 3 |
| 5.2.1. | Sprint 1..... | 3 |
| 5.2.1.1. | Sprint Planning | 3 |

| | |
|-------------------------------------|----|
| 5.2.1.2. Sprint Backlog..... | 4 |
| 5.2.1.3. Sprint Review..... | 4 |
| 5.2.1.4. Prueba de Sprint..... | 5 |
| 5.2.2. Sprint 2..... | 5 |
| 5.2.2.1. Sprint Planning | 5 |
| 5.2.2.2. Sprint Backlog..... | 6 |
| 5.2.2.3. Sprint Review..... | 6 |
| 5.2.2.4. Prueba de Sprint..... | 7 |
| 5.2.3. Sprint 3..... | 7 |
| 5.2.3.1. Sprint Planning | 7 |
| 5.2.3.2. Sprint Backlog..... | 8 |
| 5.2.3.3. Sprint Review..... | 8 |
| 5.2.3.4. Prueba de Sprint..... | 9 |
| CONCLUSIONES Y RECOMENDACIONES..... | 10 |
| Conclusiones..... | 10 |
| Recomendaciones..... | 11 |
| BIBLIOGRAFÍA | 13 |
| GLOSARIO DE TÉRMINOS..... | 20 |
| ANEXOS..... | 22 |
| Anexo A: Script..... | 22 |

Anexo B: Capturas de pantalla y demostración de la aplicación**¡Error! Marcador no definido.**

ÍNDICE DE FIGURAS Y TABLAS

ÍNDICE DE FIGURAS

| | | |
|-----------|--|----|
| Figura 1 | <i>Propuesta de método SIG con la metodología de desarrollo ágil</i> | 31 |
| Figura 2 | <i>Diagrama General de Casos de Uso</i> | 51 |
| Figura 3 | <i>Diagrama de Caso de Uso Siguiete Nivel - Gestionar Menú</i> | 52 |
| Figura 4 | <i>Diagrama de Caso de Uso A Detalle - Ingreso de Menú</i> | 53 |
| Figura 5 | <i>Diagrama de Caso de Uso A Detalle - Modificación de Menú</i> | 54 |
| Figura 6 | <i>Diagrama de Caso de Uso A Detalle – Dar de baja del Menú</i> | 55 |
| Figura 7 | <i>Diagrama de Caso de Uso A Detalle - Consulta de Menú por Nombre</i> | 56 |
| Figura 8 | <i>Diagrama de Caso de Uso Siguiete Nivel - Gestionar Pedido</i> | 58 |
| Figura 9 | <i>Diagrama de Caso de Uso A Detalle - Ingreso de Pedido</i> | 59 |
| Figura 10 | <i>Diagrama de Caso de Uso A Detalle - Modificación de Pedido</i> | 60 |
| Figura 11 | <i>Diagrama de Caso de Uso A Detalle - Eliminación de Pedido</i> | 61 |
| Figura 12 | <i>Diagrama de Caso de Uso A Detalle - Consulta de Pedidos por Cédula</i> | 63 |
| Figura 13 | <i>Diagrama de Caso de Uso Siguiete Nivel - Gestionar Reservas</i> | 66 |
| Figura 14 | <i>Diagrama de Caso de Uso A Detalle - Ingreso de Reservas</i> | 67 |
| Figura 15 | <i>Diagrama de Caso de Uso A Detalle - Modificación de Reserva</i> | 68 |
| Figura 16 | <i>Diagrama de Caso de Uso A Detalle - Eliminación de Reserva</i> | 69 |
| Figura 17 | <i>Diagrama de Caso de Uso A Detalle - Consulta de Reservas por Cédula</i> | 70 |
| Figura 18 | <i>Diagrama de Caso de Uso A Detalle - Consulta de Reservas por Nombre</i> | 72 |
| Figura 19 | <i>Diagrama de Caso de Uso Siguiete Nivel - Gestionar Inventario</i> | 74 |
| Figura 20 | <i>Diagrama de Caso de Uso A Detalle - Ingreso de Inventario</i> | 75 |

| | | |
|-----------|---|--------------------------------------|
| Figura 21 | <i>Diagrama de Caso de Uso A Detalle - Modificación de Inventario</i> | 76 |
| Figura 22 | <i>Diagrama de Caso de Uso A Detalle - Modificación de Inventario</i> | 77 |
| Figura 23 | <i>Diagrama de Caso de Uso A Detalle - Consulta de Inventario por Nombre</i> | 78 |
| Figura 24 | <i>Diagrama de Caso de Uso A Detalle - Consulta de Inventario por Categoría</i> | ¡Error! Marcador no definido. |
| Figura 25 | <i>Diagrama de Caso de Uso Siguiete Nivel - Gestionar Empleado</i> | 81 |
| Figura 26 | <i>Diagrama de Caso de Uso A Detalle - Ingreso de Empleado</i> | 82 |
| Figura 27 | <i>Diagrama de Caso de Uso A Detalle - Modificación de Empleados</i> | 83 |
| Figura 28 | <i>Diagrama de Caso de Uso A Detalle - Eliminación de Empleado</i> | 84 |
| Figura 29 | <i>Diagrama de Caso de Uso A Detalle - Consulta de Empleados por Cédula</i> | 85 |
| Figura 30 | <i>Diagrama de Caso de Uso A Detalle - Consulta de Empleado por Nombre</i> | 86 |
| Figura 31 | <i>Diagrama de Caso de Uso Siguiete Nivel - Gestionar Cliente</i> | 88 |
| Figura 32 | <i>Diagrama de Caso de Uso A Detalle - Ingreso de Cliente</i> | 89 |
| Figura 33 | <i>Diagrama de Caso de Uso A Detalle - Modificación de Cliente</i> | 90 |
| Figura 34 | <i>Diagrama de Caso de Uso A Detalle - Eliminación de Cliente</i> | 91 |
| Figura 35 | <i>Diagrama de Caso de Uso A Detalle - Consulta de Clientes por Cédula</i> | 92 |
| Figura 36 | <i>Diagrama de Caso de Uso A Detalle - Consulta de Clientes por Nombre</i> | 93 |
| Figura 37 | <i>Diagrama de secuencia crear plato</i> | 95 |
| Figura 38 | <i>Diagrama de secuencia modificar plato</i> | 96 |
| Figura 39 | <i>Diagrama de secuencia eliminar plato</i> | 97 |
| Figura 40 | <i>Diagrama de secuencia buscar plato por código</i> | 98 |
| Figura 41 | <i>Diagrama de secuencia buscar plato por nombre</i> | 99 |

| | | |
|-----------|--|-----|
| Figura 42 | <i>Diagrama de secuencia crear pedido</i> | 100 |
| Figura 43 | <i>Diagrama de secuencia modificar pedido</i> | 101 |
| Figura 44 | <i>Diagrama de secuencia eliminar pedido</i> | 102 |
| Figura 45 | <i>Diagrama de secuencia buscar pedido por código</i> | 103 |
| Figura 46 | <i>Diagrama de secuencia buscar pedido por cédula</i> | 104 |
| Figura 47 | <i>Diagrama de secuencia crear reserva</i> | 105 |
| Figura 48 | <i>Diagrama de secuencia modificar reserva</i> | 106 |
| Figura 49 | <i>Diagrama de secuencia eliminar reserva</i> | 107 |
| Figura 50 | <i>Diagrama de secuencia buscar reserva por código</i> | 108 |
| Figura 51 | <i>Diagrama de secuencia buscar reserva por nombre</i> | 109 |
| Figura 52 | <i>Diagrama de secuencia crear ingrediente</i> | 110 |
| Figura 53 | <i>Diagrama de secuencia modificar ingrediente</i> | 111 |
| Figura 54 | <i>Diagrama de secuencia eliminar ingrediente</i> | 112 |
| Figura 55 | <i>Diagrama de secuencia buscar ingrediente por código</i> | 113 |
| Figura 56 | <i>Diagrama de secuencia buscar ingrediente por nombre</i> | 114 |
| Figura 57 | <i>Diagrama de secuencia crear empleado</i> | 115 |
| Figura 58 | <i>Diagrama de secuencia modificar empleado</i> | 116 |
| Figura 59 | <i>Diagrama de secuencia eliminar empleado</i> | 117 |
| Figura 60 | <i>Diagrama de secuencia buscar empleado por cédula</i> | 118 |
| Figura 61 | <i>Diagrama de secuencia buscar empleado por nombre</i> | 119 |
| Figura 62 | <i>Diagrama de secuencia crear cliente</i> | 120 |
| Figura 63 | <i>Diagrama de secuencia modificar cliente</i> | 121 |
| Figura 64 | <i>Diagrama de secuencia eliminar cliente</i> | 122 |
| Figura 65 | <i>Diagrama de secuencia buscar cliente por cédula</i> | 123 |
| Figura 66 | <i>Diagrama de secuencia buscar cliente por nombre</i> | 124 |

| | |
|--|---|
| Figura 67 <i>Diagrama conceptual del restaurante</i> | 1 |
| Figura 68 <i>Diagrama lógico del restaurante</i> | 2 |
| Figura 70 <i>Diagrama físico del restaurante</i> | 3 |

ÍNDICE DE TABLAS

| | | |
|----------|--|--------------------------------------|
| Tabla 1 | <i>Ventajas y Desventajas de las Metodologías de Desarrollo Tradicionales...</i> | 23 |
| Tabla 2 | <i>Ventajas y Desventajas de las Metodologías de Desarrollo Ágiles</i> | 25 |
| Tabla 3 | <i>Análisis, Etapas y Comparativa de las Metodologías.....</i> | 26 |
| Tabla 4 | <i>Comparación de Metodologías Ágiles</i> | 35 |
| Tabla 5 | <i>Comparación de entornos de Desarrollo Web</i> | 38 |
| Tabla 6 | <i>Tabla de Ventajas y Desventajas de la Base de Datos.....</i> | 39 |
| Tabla 7 | <i>Tabla de Ventajas y Desventajas del Repositorio.....</i> | 41 |
| Tabla 8 | <i>Roles y responsables</i> | 43 |
| Tabla 9 | <i>Identificación de requerimientos</i> | 45 |
| Tabla 10 | <i>Priorización de requerimientos</i> | 46 |
| Tabla 11 | <i>Requerimientos funcionales.....</i> | 47 |
| Tabla 12 | <i>Requerimientos no Funcionales</i> | 48 |
| Tabla 13 | <i>Prioridades de fases del proyecto.....</i> | 49 |
| Tabla 14 | <i>Priorización de las tareas del proyecto</i> | 49 |
| Tabla 15 | <i>Flujo de trabajo con actividades y entregables</i> | ¡Error! Marcador no definido. |
| Tabla 16 | <i>Entidades</i> | 2 |
| Tabla 17 | <i>Sprint Backlog</i> | 4 |
| Tabla 18 | <i>Prueba de Sprint.....</i> | 5 |
| Tabla 19 | <i>Sprint Backlog</i> | 6 |
| Tabla 20 | <i>Prueba de Sprint.....</i> | 7 |
| Tabla 21 | <i>Sprint Backlog</i> | 8 |
| Tabla 22 | <i>Prueba de Sprint.....</i> | 9 |

CAPÍTULO I: INTRODUCCIÓN

1. MARCO DE REFERENCIA

1.1. Justificación

El surgimiento de emprendimientos centrados en la industria alimentaria ha tenido un impacto significativo en la sociedad. En el Distrito Metropolitano de Quito, se observa una amplia variedad de restaurantes que ofrecen comida de distintos tipos. Sin embargo, muchos de estos establecimientos siguen gestionándose de manera tradicional y carecen de un sistema eficiente para su administración.

En la mayoría de los casos, los restaurantes realizan compras de forma improvisada, basándose en su percepción o cuando se agotan los ingredientes. Esto conlleva a una distribución ineficiente del tiempo debido a las compras frecuentes, además de una falta de control sobre los ingredientes que se encuentran agotados y los que se necesitan para el día. Por lo tanto, resulta fundamental que estos restaurantes implementen un sistema de administración de inventario para gestionar el stock de cada producto y realizar compras de acuerdo con estándares establecidos, de modo que puedan garantizar un funcionamiento fluido de su negocio en el día a día.

Otro aspecto crucial es la adecuada gestión de las ventas en los restaurantes, ya que es común que estos establecimientos registran de forma manual las transacciones generadas, y al finalizar, realizan un proceso de verificación de las ganancias obtenidas junto con los registros detallados de parte de los repartidores y encargados de caja, ocasionando problemas al no cuadrando esta información, debido a esto se desarrollará una aplicación web esto permitirá agilizar el proceso de registro de las ganancias diarias, reduciendo el tiempo necesario para conciliar la información registrada por distintos miembros del equipo de trabajo.

1.2. Planteamiento del problema

Durante la crisis provocada por la pandemia de COVID-19, se evidenció el cierre de numerosos establecimientos gastronómicos por no tener contacto con los clientes, factores como el reducir el aforo de los restaurantes fueron las principales razones de cerrar los negocios al no estar preparados para trabajar bajo reservas, aunque para restaurantes que han dado el paso a la tecnología, y han desarrollado sistemas encargados de reservas y administración permitieron un incremento en sus ganancias y un mayor posicionamiento de su marca, generando un gran crecimiento a su negocio.

Los restaurantes al no contar con un sistema de inventario no se pueden tomar decisiones adecuadas, teniendo un deficiente control sobre la compra de productos, manejo de las ganancias, e información de las reservas para garantizar un funcionamiento óptimo del negocio, resulta fundamental implementar una aplicación web destinada a la gestión de reservas y administración de restaurantes. Esta solución permitirá reducir significativamente el tiempo empleado en el control y revisión de información, brindando así beneficios tangibles para la gestión del establecimiento.

Asimismo, el aprovisionamiento del inventario representaba un desafío, dado que las compras se efectuaban únicamente cuando los ingredientes se agotaban. Esto resultaba en múltiples compras diarias o la adquisición de ingredientes que no eran prioritarios en ese momento. Para abordar esta situación, la aplicación móvil enviará notificaciones adecuadas al responsable del abastecimiento de ingredientes, de modo que pueda realizar una única compra de todos los elementos necesarios en función de los requerimientos actuales.

1.3. Objetivo General

Desarrollar una aplicación web responsiva para la gestión de reservas y administración del restaurante.

1.4. Objetivos Específicos

- Analizar los requerimientos de los restaurantes para la gestión y administración de reservas.
- Realizar el diseño del sistema de la base de datos según los requerimientos del sistema.
- Desarrollar el diseño de las interfaces de la aplicación web responsiva centrados en los usuarios.
- Diseñar y desarrollar prototipos rápidos como avances del sistema.
- Desarrollar el sistema de restaurante aplicando reglas de usabilidad.

1.5. Antecedentes

Según López et al. (2020), los 76 días de confinamiento en Ecuador ha resultado en una costosa factura para la economía del país. Según datos oficiales, se han acumulado pérdidas por USD 15 863 millones desde el 16 de marzo del hasta el año 2021. Además, una parte importante de ese costo también fue soportada por el sector manufacturero, que se estimó en un 70%.

Según estimaciones de la Empresa Pública Metropolitana de Gestión de Destinos, el 2,4% de todas las actividades productivas de la capital de la ciudad se atribuyeron a las industrias de alimentación en el primer trimestre de 2020, según (Quito Turismo, 2020).

De acuerdo con Castañeda (2016) el desarrollo de contenidos online responsables es esencial para muchas empresas, estas cuentan con sitios web de acceso público. A su vez, destaca que contar con interfaces amigables para el usuario brinda un efecto positivo, permitiendo que los usuarios disfruten de su uso y sientan comodidad al realizar acciones de forma autónoma.

1.6. Alcance

Mediante este trabajo se brindará a los restaurantes el correcto paso a la tecnología, en el cual se contempla desarrollar procesos de gestión de inventario, generando un mayor control y reaprovisionamiento de recursos necesarios para el funcionamiento diario.

A su vez, se desarrollará un control de ventas, permitiendo que los restaurantes ya no realicen esta tarea de forma manual, generando un correcto registro de los ingresos y egresos generados en la jornada de trabajo.

Otro factor importante que se desarrollará es el manejo de reservas, permitiendo trabajar bajo tiempos concretos, brindando al usuario mayor comodidad y flexibilidad.

Por último, este trabajo de titulación desarrollará y automatizará los procesos de mencionados mediante un aplicativo web, registrando la información en una base de datos, y brindando información precisa para cada uno de nuestros roles (usuario, caja, inventario y repartidor).

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

2. Marco Teórico

2.1. Página Web

Una página web es un documento o recurso digital accesible a través de Internet que contiene información multimedia, como texto, imágenes, videos y enlaces. Las páginas web son creadas utilizando lenguajes de marcado como HTML, CSS y JavaScript, y pueden ser accedidas mediante un navegador web. Se realiza en un formato estandarizado que proporciona una base sólida para mostrar y organizar información en la web, utilizando un lenguaje común y herramientas de desarrollo que garantizan la coherencia y la compatibilidad en la presentación de contenido en Internet (Briones Chauca, 2019).

2.2. HTML

El lenguaje de desarrollo de páginas web es un sistema compuesto por etiquetas que permiten dar formato y estructura a la visualización de contenido en la pantalla. Está conformado por una sección de cabecera y una sección de cuerpo en el documento. El lenguaje estándar llamado HTML es utilizado para presentar diversa información, como texto, imágenes y videos, en la red más extensa de todo Internet. Este formato se basa en un lenguaje que está compuesto por plantillas CSS y otras herramientas de desarrollo, permitiendo así una estructura y diseño coherentes en la presentación de contenido en línea (Briones Chauca, 2019).

2.3. CCS

El CSS es un lenguaje de estilos ampliamente utilizado que tiene como objetivo principal definir la presentación, el formato y la apariencia de documentos de marcaje, tales como HTML, XML u otros similares. Su función principal consiste en aplicar estilos

visuales a los elementos contenidos en estos documentos (Guapi Auquilla, 2018). De acuerdo con el World Wide Web Consortium (W3C), el CSS es un lenguaje que maneja hojas de estilo el cual es utilizado para realizar la presentación de un documento estructurado escrito en HTML, XML u otro lenguaje de marcado equivalente (W3C, 2013). Su propósito es separar la estructura y el contenido del documento de su presentación visual, lo que permite un mayor control y flexibilidad en la apariencia de las páginas web. Además, según Mozilla Developer Network (MDN) Web Docs, el CSS se emplea para precisar la apariencia visual de una página web, agregando elementos como tipografía, colores, diseño, espaciado, y efectos visuales (MDN Web Docs, 2023). Esto permite personalizar y dar estilo a los elementos de una página, mejorando así la experiencia visual de los usuarios.

2.4. Javascript

JavaScript es ampliamente reconocido como uno de los lenguajes de programación más populares en la actualidad. Su principal ventaja radica en ser un lenguaje interpretado, lo que significa que el código no requiere ser compilado previamente, sino que puede ser ejecutado directamente en un navegador sin necesidad de procesos intermedios (Anchundia Medrano, 2022). Según Mozilla Developer Network (MDN) Web Docs (2023) JavaScript es un "lenguaje de programación interpretado, de alto nivel y de múltiples paradigmas". El cual es ampliamente utilizado para agregar interactividad y dinamismo a las páginas web, permitiendo la manipulación de elementos HTML, la gestión de eventos y la comunicación con servidores.

Además, en el libro "JavaScript: la guía definitiva" de David Flanagan (2020), se define JavaScript como un lenguaje de scripting que se ejecuta en el navegador web

del cliente y que brinda a los desarrolladores la capacidad de crear contenido interactivo y dinámico. Su interpretación directa en el navegador brinda flexibilidad y agilidad en el desarrollo de aplicaciones web.

2.5. PHP

PHP significa "Preprocesador de hipertexto", es un lenguaje de programación de código abierto ampliamente utilizado para el desarrollo web. Fue creado originalmente por Rasmus Lerdorf en 1994 y ha evolucionado a lo largo de los años hasta convertirse en uno de los lenguajes más populares para la creación de aplicaciones web dinámicas (PHP, 2023). PHP es un lenguaje interpretado y se ejecuta en el lado del servidor, lo que significa que el código PHP se procesa en el servidor antes de que se envíe el resultado al cliente. Esto permite generar contenido dinámico en tiempo real, como páginas web personalizadas, formularios interactivos y sistemas de gestión de bases de datos.

PHP tiene una amplia compatibilidad con una amplia gama de sistemas operativos (Windows, Linux, macOS, etc.) y servidores web (Apache, Nginx, IIS, etc.). Además, es compatible con una variedad de bases de datos, como MySQL, PostgreSQL y SQLite. PHP se integra de manera natural con HTML, lo que permite incrustar código PHP directamente en archivos HTML para mezclar contenido estático con contenido dinámico.

PHP cuenta con una amplia variedad de bibliotecas y frameworks populares, como Laravel, Symfony y CodeIgniter, que simplifican el desarrollo web y brindan una estructura organizada para construir aplicaciones más robustas y escalables. PHP está diseñado específicamente para el desarrollo web y proporciona funciones integradas

para trabajar con formularios, cookies, sesiones, archivos, bases de datos y más. Esto facilita la creación de aplicaciones web completas y funcionales.

2.6. Arquitectura de Software

La arquitectura de software implica la creación de abstracciones, la descomposición y composición de elementos, así como el enfoque en estilos y estética. También está relacionada con el diseño e implementación de la estructura de alto nivel del software. Los arquitectos de software construyen esta arquitectura utilizando una variedad de elementos arquitectónicos cuidadosamente seleccionados. Estos elementos cumplen con la mayoría de los requisitos funcionales y de rendimiento del sistema, así como con otros requisitos no funcionales como confiabilidad, escalabilidad, portabilidad y disponibilidad del sistema (Kruchten, 2015).

Según la definición de la IEEE Sociedad informática (2020), la arquitectura de software es "la estructura fundamental de un sistema de software, los componentes y las relaciones entre ellos, y los principios y las directrices que gobiernan su diseño y evolución a lo largo del tiempo". Esta definición destaca la importancia de la estructura y las relaciones entre los componentes en el diseño y evolución de sistemas de software. Además, en el libro "Arquitectura de software en la práctica" de los autores Bass et al. (2012), se define la arquitectura de software como la estructura del sistema, que consiste en elementos de software, las relaciones entre ellos y los atributos de estos elementos". En donde también se hace énfasis en la importancia de satisfacer requisitos tanto funcionales como no funcionales en el diseño arquitectónico.

2.6.1. Modelo Vista Controlador (MVC)

El modelo vista controlador (MVC) se ha establecido como la arquitectura estándar en numerosos lenguajes de programación orientados a objetos, especialmente en proyectos que se centran en el desarrollo de interfaces de usuario. Su funcionamiento se basa en la separación del código en tres capas independientes, cada una con responsabilidades definidas, conocidas como modelos, vistas y controladores (Alonso-Aranda, 2019).

Según el libro "Patrones de diseño: elementos del software orientado a objetos reutilizable" de los autores Gamma et al. (1994) señalan que el modelo vista controlador (MVC) es un patrón arquitectónico que promueve la separación de preocupaciones al dividir una aplicación en tres componentes principales: el modelo, que representa los datos y la lógica de negocio; la vista, que se encarga de la presentación de la información al usuario; y el controlador, que maneja las interacciones y coordina las acciones entre el modelo y la vista. Además, en el artículo "Patrón arquitectónico MVC (modelo-vista-controlador)" de Microsoft, destaca que el objetivo del MVC es lograr la reutilización de código, la modularidad y la facilidad de mantenimiento al separar las preocupaciones en componentes independientes y definidos claramente (Microsoft, 2022).

2.7. Metodología de Aplicada

La metodología de investigación aplicada se refiere a un proceso que tiene como objetivo convertir el conocimiento teórico derivado de la investigación básica en conceptos, prototipos y productos de aplicación práctica. Este enfoque busca poner en práctica los resultados de la investigación teórica para generar soluciones concretas y aplicables (Lozada, 2014, p. 5).

En el libro "Metodología de la investigación: Enfoque cualitativo y cuantitativo" de los autores Hernández et al. (2014), se define la investigación aplicada como un tipo de investigación que se enfoca en resolver problemas prácticos y generar conocimiento que pueda ser aplicado en la práctica. Su objetivo es producir resultados que sean útiles y relevantes para el contexto real.

Además, en el artículo "Metodología de investigación: una guía paso a paso para principiantes" del autor Kumar (2019), destaca que la investigación aplicada se centra en la aplicación de los resultados de la investigación básica en contextos prácticos, con el fin de abordar problemas concretos y generar soluciones prácticas.

2.8. Metodología Desarrollo de Software

La metodología de desarrollo de software se refiere a un conjunto de procesos sistemáticos que se siguen para diseñar, implementar y mantener un producto de software, desde el inicio de la necesidad del producto hasta la consecución de los objetivos para los cuales fue creado (Maida & Pacienza, 2015).

2.8.1. SCRUM

SCRUM es una metodología que combina buenas prácticas y trabajo colaborativo de equipo para lograr resultados óptimos, a través de la colaboración de un equipo altamente competente. En SCRUM, se realizan entregas incrementales del proyecto, las cuales son priorizadas según su contribución al proyecto y valoradas por los usuarios finales del mismo (Ramírez et al., 2018, p. 1068).

En el libro "Gestión ágil de proyectos con Scrum" del autor Schwaber (2004), se describe SCRUM como un marco de trabajo ágil que se centra en la colaboración, la adaptabilidad y la entrega de valor en forma iterativa e incremental. Se enfoca en la autogestión del equipo y la entrega continua de funcionalidades, con el objetivo de satisfacer las necesidades cambiantes del cliente de manera flexible y efectiva.

Además, en el artículo " Una descripción general de la metodología Scrum en el desarrollo ágil" de los autores Sutherland y Schwaber (2017), destacan los principios clave de SCRUM, como la transparencia, la inspección y la adaptación. SCRUM promueve la comunicación constante entre el equipo y los stakeholders, así como la mejora basada en el aprendizaje y la retroalimentación.

2.9. UML

UML (Lenguaje de modelado unificado) es un lenguaje que brinda la capacidad de modelar, construir y documentar los diversos elementos que componen un sistema de software orientado a objetos. Con el tiempo, UML se ha convertido en el estándar ampliamente aceptado y utilizado en la industria para el desarrollo y diseño de sistemas de software(Grau & Segura, 2000).

Además, en el artículo "Una revisión sistemática sobre la aplicación de UML para modelar sistemas de software" de los autores Ali y Shallal (2019), se destaca que UML proporciona un conjunto de diagramas y notaciones que permiten a los desarrolladores capturar y comunicar de manera efectiva los requisitos, la estructura y el comportamiento de los sistemas de software. Se utiliza ampliamente en la

industria para mejorar la comprensión, la colaboración y la calidad en el desarrollo de software

2.10. Base de Datos

Las bases de datos son sistemas informáticos diseñados para almacenar y administrar datos, ofreciendo soluciones a los desafíos asociados con la gestión de la información. A lo largo de los años, las bases de datos han evolucionado y se han convertido en herramientas fundamentales en diversas aplicaciones y entornos (López Herrera, 2016)

Según el libro "Sistemas de bases de datos: diseño, implementación y gestión" de los autores Coronel y Morris (2016), se define a una base de datos como un conjunto organizado de datos relacionados y estructurados, que se almacenan y se pueden acceder electrónicamente. Proporcionando de esta manera una forma eficiente de almacenar, recuperar, modificar y administrar datos, brindando ventajas como el ahorro de espacio, la rápida recuperación de información y la reducción de la necesidad de manipulación de documentos físicos.

Además, en el artículo "Sistemas de gestión de bases de datos: una revisión" de los autores Gupta y Chauhan (2015), destacan que las bases de datos permiten la gestión centralizada de datos, lo que garantiza la integridad y consistencia de la información. También señalan que facilitan el acceso rápido a los datos mediante consultas y ofrecen funciones de seguridad para proteger la confidencialidad y la integridad de los datos almacenados.

2.10.1. Gestor de Base de Datos

Los gestores de bases de datos (SGBD) brindan a los usuarios la capacidad de definir los datos que se almacenarán utilizando un modelo de datos. Es importante destacar que la mayoría de los sistemas de gestión de bases de datos actuales se basan en el modelo relacional de datos (López Herrera, 2016).

Según el libro "Sistemas de bases de datos: El libro completo" de los autores García et al. (2021), un gestor de bases de datos es un software que permite a los usuarios definir, crear, mantener y manipular bases de datos. Proporcionando una interfaz para interactuar con la base de datos, ofreciendo funcionalidades como el almacenamiento eficiente de datos, la administración de transacciones y la ejecución de consultas.

Además, en el artículo "Una encuesta sobre los sistemas de gestión de bases de datos" de los autores Singh y Chauhan (2021), resaltan que los SGBD desempeñan un papel crucial en la organización y el acceso a grandes volúmenes de datos. Estos sistemas permiten la definición de esquemas de datos, la manipulación de información y la implementación de políticas de seguridad y respaldo para garantizar la integridad y disponibilidad de los datos.

2.10.2. Base de Datos Relacional

Una base de datos relacional es un tipo de sistema de almacenamiento y acceso a datos que permite la gestión de puntos de datos interrelacionados. Este tipo de bases de datos se fundamenta en el modelo relacional, el cual proporciona una forma clara y estructurada de representar la información en forma de tablas.

Según el libro "Sistemas de gestión de bases de datos" de los autores Ramakrishnan y Gehrke (2020), definen a la base de datos relacional como un sistema de gestión de bases de datos que organiza y almacena datos en forma de tablas, donde cada tabla representa una entidad o relación en el mundo real. Estas tablas están interconectadas a través de claves primarias y claves foráneas, permitiendo establecer relaciones y realizar consultas para acceder a la información.

Además, en el libro "Bases de datos relacionales: conceptos, diseño e implementación" de los autores Elmasri y Navathe (2019), enfatizan que las bases de datos relacionales han sido ampliamente adoptadas en la industria debido a su flexibilidad, facilidad de uso y capacidad para representar relaciones complejas entre los datos. Su modelo basado en tablas y relaciones permite una organización lógica y eficiente de la información, lo que facilita la manipulación y recuperación de datos.

2.10.3. MYSQL

MySQL es ampliamente reconocido como uno de los sistemas de gestión de bases de datos más utilizados para aplicaciones web. Es una solución de código abierto que goza de gran popularidad debido a su versatilidad y a las continuas actualizaciones que recibe, incorporando constantemente nuevas características y mejoras en cuanto a seguridad (Lozano Banqueri, 2018).

Según el sitio web oficial de MySQL (2023), MySQL es un sistema de gestión de bases de datos relacional de código abierto que brinda una plataforma sólida y confiable para el almacenamiento y acceso eficiente de datos. Es especialmente conocido por su capacidad para respaldar aplicaciones web y su facilidad de uso, lo

que lo convierte en una elección popular entre desarrolladores y administradores de bases de datos.

Además, en el artículo "Un estudio comparativo de los sistemas de gestión de bases de datos relacionales" de los autores Bhagwat y Patil (2021), se destaca la importancia de MySQL en el ámbito de las bases de datos, mencionando su amplia adopción y su capacidad para gestionar grandes volúmenes de datos con buen rendimiento y eficiencia.

2.11. IDE

Un IDE es una herramienta de software que proporciona a los desarrolladores un conjunto de características y herramientas integradas para facilitar el desarrollo de software (Álvarez, 2017). Un IDE generalmente incluye un editor de código fuente, un depurador, un compilador o intérprete, herramientas de construcción y un entorno de ejecución. También puede ofrecer características adicionales como resaltado de sintaxis, completado automático de código, depuración paso a paso, refactorización de código, control de versiones, administración de proyectos y plugins o extensiones para ampliar su funcionalidad.

2.12. Diagrama

Un diagrama general es una representación visual de alto nivel de un sistema o proceso que muestra sus componentes principales y las principales entre ellos. Los diagramas generales se realizan para realizar una representación visual simplificada de un sistema o proceso complejo, con el fin de que ayude a comprender su estructura y funcionamiento de manera más clara y concisa. Estos diagramas pueden incluir

diferentes elementos y artefactos dependiendo del propósito y contexto del sistema que se está representando.

2.12.1. Diagrama lógico

El diagrama lógico representa la estructura y las relaciones de los datos en un nivel abstracto (Coronel & Morris, 2018). En este tipo de diagrama se incluyen todas las entidades y las relaciones entre ellas, se especifican todos los atributos de cada entidad, se establece la clave principal de cada entidad y se definen las claves externas que identifican las relaciones entre diferentes entidades. Además, en este nivel se realiza la normalización de los datos. El diagrama de datos lógicos proporciona una descripción completa y detallada de los datos y su estructura, sin considerar los aspectos físicos de su implementación en la base de datos

2.12.2. Diagrama físico

El diagrama físico muestra cómo se implementan los datos en un sistema de gestión de bases de datos específico (Coronel & Morris, 2018). El diagrama físico representa la construcción real del modelo en la base de datos. Muestra todas las estructuras de tabla, incluyendo nombres de columnas, tipos de datos, restricciones, claves primarias, claves externas y relaciones entre las tablas. En este diagrama, se especifican todas las tablas y columnas, se utilizan claves externas para identificar las relaciones entre las tablas, y la desnormalización puede ocurrir según los requisitos del usuario. Es importante tener en cuenta que el modelo de datos físicos puede ser diferente del modelo de datos lógicos debido a consideraciones físicas. Los pasos básicos para diseñar el modelo de datos físicos son convertir entidades en tablas,

convertir relaciones en claves externas, convertir atributos en columnas y realizar modificaciones según las restricciones o requisitos físicos.

2.12.3. Diagrama conceptual

El diseño conceptual de bases de datos es la etapa que sigue al análisis de requisitos en el proceso de diseño de una base de datos (Casas, 2021). Su objetivo es crear un esquema conceptual de alto nivel e independiente de la tecnología de implementación, basándose en las especificaciones y requisitos de un problema del mundo real. Durante esta etapa, se extraen las necesidades y requisitos del problema para sintetizar un modelo visual que represente los datos y las restricciones conceptuales que se desean en el sistema de información. Un diagrama conceptual, también conocido como diagrama de entidad-relación, es una representación visual que muestra las entidades y las relaciones entre ellas en un sistema o dominio específico.

En el diseño conceptual, se establece una estructura de información independiente de la tecnología específica. No se considera qué tipo de base de datos se utilizará (relacional, orientada a objetos, etc.), ni el sistema de gestión de bases de datos o el lenguaje de programación concreto. El enfoque principal se centra en la estructura de la información, dejando las cuestiones tecnológicas para resolver en etapas posteriores del diseño.

2.12.4. Diagrama de clases

Los diagramas de clases son representaciones estáticas que describen la estructura de un sistema mediante la visualización de las clases y sus atributos (Casas, 2021). Estos diagramas desempeñan un papel fundamental en el desarrollo orientado

a objetos. El diagrama de clases muestra toda la estructura estática de un sistema, representando las clases, sus atributos, métodos y las relaciones entre ellas conocidas como asociaciones en UML.

2.12.5. Diagramas de casos de uso

Un diagrama de casos de uso es una representación gráfica que muestra las interacciones entre los actores (usuarios o sistemas externos) y el sistema en estudio, identificando los diferentes casos de uso y las relaciones entre ellos (Fowler, 2019). El diagrama de casos de uso es una representación en el lenguaje de modelado unificado (UML) que permite visualizar los procesos empresariales, sistemas y procesos de programación orientada a objetos. A diferencia de un lenguaje de programación, UML es un lenguaje de modelado estandarizado utilizado para representar sistemas existentes o planeados. En este tipo de diagrama, se organiza y se establecen relaciones entre los objetos involucrados en el sistema.

2.12.6. Diagrama de secuencia

Los diagramas de secuencia son representaciones visuales que describen las interacciones entre diferentes objetos a lo largo del tiempo. Estos diagramas se centran en mostrar el flujo temporal de mensajes que se envían y reciben entre los objetos involucrados (Casas, 2021). Proporcionan una representación detallada de cómo los objetos colaboran entre sí para lograr un determinado resultado. Los diagramas de secuencia son ampliamente utilizados en el análisis y diseño de sistemas, ya que permiten comprender y visualizar el comportamiento dinámico de un sistema en términos de las interacciones entre sus componentes. Estos diagramas ayudan a identificar las acciones que ocurren en secuencia y cómo los objetos se comunican

entre sí a medida que se desarrolla una determinada funcionalidad o escenario. Al capturar el flujo temporal de los mensajes, los diagramas de secuencia facilitan la comprensión de la lógica y la dinámica del sistema, lo que resulta valioso para la identificación de errores, la optimización del diseño y la comunicación efectiva entre los miembros del equipo de desarrollo.

CAPÍTULO III: METODOLOGÍA

3. Metodología de desarrollo de la tesis

Para el desarrollo del presente trabajo de titulación se aplica dos fases: metodología de investigación y metodología de desarrollo.

3.1. Metodología Aplicada

La metodología aplicada se refiere a un enfoque de investigación que tiene como objetivo principal generar conocimiento con una aplicación directa en la práctica y en la solución de problemas reales en diversos campos. A diferencia de la investigación básica, que busca ampliar el conocimiento teórico sin un enfoque específico en la aplicación práctica, la metodología aplicada se centra en producir resultados y soluciones concretas que puedan ser implementadas en la sociedad.

Según la "Enciclopedia de diseño de investigación" del autor Salkind (2010), se define a la metodología aplicada por la que se caracteriza por la utilización de métodos y técnicas de investigación que permiten abordar problemas y desafíos del mundo real. Se basa en la combinación de conocimientos teóricos con la experiencia práctica para desarrollar intervenciones, programas o políticas que tengan un impacto directo y positivo en la sociedad.

Además, en el artículo "Investigación Aplicada: Antecedentes Históricos y Tendencias Actuales" de los autores Grimm y Hans (2018), se destaca la importancia de la metodología aplicada en la transferencia de conocimiento y tecnología desde el ámbito académico hacia la práctica. Se resalta la necesidad de colaboración entre investigadores, profesionales y comunidades para asegurar que la investigación aplicada aborde problemas relevantes y genere resultados significativos.

La metodología aplicada en este sentido tiene como objetivo resolver un determinado problema, centrándose en la búsqueda de conocimiento para su aplicación, busca por medio del conocimiento generado de la investigación generar un valor elevado para el público objetivo, por la creación de nuevos productos.

3.2. Metodología para el Desarrollo de Software

En la parte investigativa se analizan los requerimientos de los participantes involucrados, así como los requerimientos y objetivos de un restaurante. Para que el presente trabajo cumpla los objetivos previamente definidos, se requiere de una metodología de desarrollo de software para la aplicación web. En este capítulo se presentan dos tipos de metodologías: metodologías tradicionales y metodologías ágiles. Posteriormente, se realiza una comparativa y se concluye cuál de los dos tipos de metodologías es la que mejor se adapta al propósito del trabajo de titulación.

La metodología para el desarrollo de software es un conjunto de enfoques, técnicas y prácticas que se utilizan para gestionar y guiar el proceso de creación de software. Estas metodologías están diseñadas para mejorar la calidad de los productos y optimizar los procesos de desarrollo. Según Velásquez et al. (2019), el uso de metodologías formales y estándares es fundamental para controlar y asegurar la eficiencia de los procesos de desarrollo de software.

En el artículo "Metodologías de desarrollo de software: una revisión" de Azeem y Hussain (2020), se destaca la importancia de las metodologías de desarrollo de software en la mejora de la calidad de los productos y la eficiencia de los procesos. Se menciona que estas metodologías proporcionan un marco estructurado para el desarrollo de software, ayudando a los equipos a gestionar los recursos, establecer plazos y garantizar la entrega de productos de alta calidad.

Además, en el libro "Ingeniería de software: el enfoque de un profesional" de los autores Pressman y Maxim (2020), se aborda el tema de las metodologías para el desarrollo de software, resaltando su importancia para establecer un enfoque sistemático y controlado en el proceso de creación de software. Se menciona que estas metodologías proporcionan directrices y prácticas para garantizar la entrega exitosa de productos software.

3.2.1. Metodologías de Desarrollo Tradicionales

Las metodologías tienen como propósito diseñar soluciones de software informático y permiten organizar los equipos de trabajo, optimizando su desarrollo y gestionando cada etapa y encargado de este, reduciendo la dificultad al agilizar el proceso y garantizando mejores resultados. Las metodologías de desarrollo tradicionales se caracterizan por definir los objetivos y requisitos al inicio del proyecto, generando ciclos de desarrollo inflexibles.

Las metodologías de desarrollo tradicionales se caracterizan por concebir el proyecto como un todo de grandes dimensiones y estructura definida. Estas metodologías siguen un enfoque secuencial y lineal, sin posibilidad de retroceso, donde el proceso se considera rígido y no sujeto a cambios. Los requerimientos del proyecto se acuerdan de antemano y se espera que se mantengan constantes a lo largo del proceso de desarrollo. Esto implica una planificación previa extensa y una comunicación limitada con el cliente una vez que se ha finalizado dicha planificación (Montero et al., 2018).

En el artículo "Metodologías ágiles vs tradicionales: un estudio comparativo" de los autores Vijayasarithi y Dhandapani (2019), se realiza un estudio comparativo entre las metodologías ágiles y las metodologías tradicionales de desarrollo. Se discute que, a diferencia de las metodologías tradicionales, las metodologías ágiles promueven la flexibilidad, la colaboración y la adaptabilidad en el proceso de desarrollo, permitiendo una mayor

interacción con el cliente y una respuesta más ágil a los cambios en los requerimientos del proyecto.

Además, en el libro "Ingeniería de software: el enfoque de un profesional" de los autores Pressman y Maxim (2020), se examinan las metodologías tradicionales de desarrollo de software. Se señala que estas metodologías se basan en una planificación y documentación exhaustiva antes de comenzar el desarrollo, con un enfoque lineal y secuencial. Sin embargo, también se destaca que este enfoque puede ser inflexible y poco adecuado para proyectos donde los requerimientos son cambiantes o poco claros.

Tabla 1

Ventajas y Desventajas de las Metodologías de Desarrollo Tradicionales

| Ventajas | Desventajas |
|--|--|
| <ul style="list-style-type: none">• Organización en los equipos de trabajo.• Garantiza resultados de calidad.• Trabajan en base a una visión clara, basándose en objetivos para el cumplimiento de requisitos. | <ul style="list-style-type: none">• Inflexible ante cambios.• Poca comunicación con el usuario.• Procesos lineales que no permiten cambios una vez definidos los requerimientos.• Requiere mayor tiempo de desarrollo del proyecto ya que no se puede avanzar hasta culminar la etapa previa. |

Fuente: Autor: Cadena, S. (2023)

3.2.2. Metodologías de Desarrollo Ágiles

En la actualidad, las empresas se encuentran inmersas en un entorno global que experimenta cambios constantes. Esto implica la necesidad de adaptarse rápidamente a nuevas oportunidades, mercados en evolución y condiciones económicas cambiantes. Además, se enfrentan al desafío de responder al surgimiento de productos y servicios nuevos y altamente competitivos (Ignacio & Paola, 2015).

En el artículo "Desarrollo ágil de software: un estudio empírico de desafíos y beneficios" de los autores Al-Daraiseh y Samarah (2019), se realiza un estudio empírico sobre los desafíos y beneficios de la metodología de desarrollo ágil. Se destaca que las metodologías ágiles permiten a las empresas responder de manera más eficiente y efectiva a los cambios en el entorno empresarial, fomentando la flexibilidad, la colaboración y la adaptabilidad en el desarrollo de software.

Además, en el libro "Gestión ágil de proyectos: creación de productos innovadores" del autor Highsmith (2019), se explora en profundidad el enfoque ágil en la gestión de proyectos. Se resalta que las metodologías ágiles se centran en la entrega rápida y continua de valor al cliente, permitiendo una mayor interacción y retroalimentación con los stakeholders a lo largo del proceso de desarrollo.

En este sentido, las metodologías ágiles se caracterizan por su flexibilidad y adaptabilidad. Su desarrollo permite dividir proyectos en partes reducidas, lo cual es factible porque se incluye al usuario en la comunicación y desarrollo del proyecto, permitiendo realizar cambios y revisiones oportunas. De esta forma, se asegura una retroalimentación constante en el cumplimiento de los objetivos y requerimientos.

Tabla 2

Ventajas y Desventajas de las Metodologías de Desarrollo Ágiles

| Ventajas | Desventajas |
|--|---|
| <ul style="list-style-type: none"> • Adaptabilidad y flexibilidad ante los cambios. • Constante comunicación con el usuario. • Entregas periódicas de los avances del proyecto. • Constante revisión y supervisión de cada etapa del proyecto. | <ul style="list-style-type: none"> • Requiere compromiso de los líderes e integrantes del equipo de trabajo. • Los objetivos y requerimientos pueden variar, generando retrasos en los tiempos establecidos de entrega. • Puede requerir más recursos en caso de retrasos o errores presentados en las diferentes etapas del proyecto. |

Autor: Cadena, S. (2023)

3.3. Análisis, Etapas y Comparativa de las Metodologías

Las metodologías de desarrollo tradicionales y ágiles disponen de sus propias características y beneficios dependiendo del enfoque del proyecto. Por esta razón, se presentarán puntos importantes para definir qué metodología se adapta más a los propósitos del presente tema de titulación:

1. **Arquitectura:** Define cómo está estructurada la arquitectura de software. Estas metodologías promueven cómo se desarrolla un proyecto y, en base a sus requerimientos, pueden aplicar arquitecturas modulares o flexibles a cambios.
2. **Ciclo de Vida:** Son las fases para el desarrollo del proyecto y pueden ser iterativas o secuenciales según la metodología.

3. **Documentación:** Determina el enfoque que se dará a la documentación. Esta puede requerir, según el proyecto, una documentación detallada que se enfoca en los procesos del proyecto, o una documentación reducida enfocada en los resultados.
4. **Duración:** La duración es el tiempo que llevará en culminar el proyecto, aunque dependerá de los objetivos, requerimientos y tiempos de entrega.
5. **Equipo de trabajo:** Es la cantidad de integrantes que tiene un grupo.
6. **Comunicación con el cliente:** Trata de definir si requiere la participación del cliente para el desarrollo.
7. **Retroalimentación:** Es el proceso para recibir comentarios respecto a revisiones de las fases del proyecto que permitan realizar un cambio o una acción oportuna.

En base a los 7 puntos mencionados, se definirá una tabla comparativa para concluir qué metodología cumple con estas características para la elaboración del trabajo de titulación.

Tabla 3

Análisis, Etapas y Comparativa de las Metodologías

| Puntos de Importantes | Metodología Tradicional | Metodología Ágil | Explicación |
|-----------------------|-------------------------|------------------|---|
| Arquitectura | | X | Al no tener un caso de estudio como referencia, no se puede trabajar de forma detallada. Se escogió la Metodología Ágil porque brinda tolerancia a los cambios presentados en el desarrollo y se caracteriza por ser adaptable a los cambios. |
| Ciclo de Vida | | X | Al definir que la metodología de investigación requiere de |

| | | |
|--------------------------|----------|---|
| | | encuestas y entrevistas para generar información, el ciclo de vida de las Metodologías Ágiles se adapta más a los requerimientos del proyecto, ya que suele tener un proceso iterativo. A diferencia de las metodologías tradicionales, que son secuenciales, siendo una limitante al no tener información y procesos a detalle, lo que impide el progreso secuencial del proyecto. |
| Documentación | X | La metodología tradicional se caracteriza por requerir documentación detallada de procesos e información, a diferencia de la metodología ágil que se caracteriza por su documentación escasa o enfocada en los resultados. Esto la hace más beneficiosa para este proyecto al no disponer de información detallada. |
| Duración | X | La duración de la metodología tradicional suele estar enfocada en proyectos de mediano/largo plazo, mientras que la metodología ágil es enfocada en proyectos de corta duración, siendo ideal para la elaboración del presente proyecto. |
| Equipo de trabajo | X | La metodología tradicional suele requerir un mayor número de integrantes en el equipo de trabajo, mientras que la |

| | | |
|------------------------------------|----------|---|
| | | metodología ágil trabaja con grupos pequeños de hasta 10 integrantes. |
| Comunicación con el cliente | X | La metodología ágil es indiscutiblemente requerida en esta fase, ya que, al no contar con un caso de estudio específico ni información, se requiere generar dicha información por medio de entrevistas y encuestas a clientes o grupos objetivos para el desarrollo del proyecto. |
| Retroalimentación | X | Al requerir retroalimentación constante del cliente o grupo objetivo, se requieren evaluaciones y revisiones periódicas, siendo la metodología ágil la que más se adapta a estas necesidades. |

Autor: Cadena, S. (2023)

En la tabla 3 se presenta un cuadro comparativo de las metodologías ágiles y tradicionales en el cual se definen los puntos importantes para determinar qué metodología se adapta más a las necesidades del proyecto. Por lo tanto, podemos concluir que la metodología ágil cumple con los requisitos y se adapta a las necesidades del trabajo de titulación.

3.4. Metodologías Ágiles de Desarrollo

Las Metodologías Ágiles de Desarrollo disponen de características propias y beneficios dependiendo el enfoque del proyecto, por esta razón, se detalla puntos importantes para definir que Metodologías Ágiles de Desarrollo se adapta más a los propósitos del presente tema de titulación:

- **Aprendizaje:** Se requiere de metodologías ágiles sencillas de dominar y comprender, que permitan solucionar requerimientos de forma rápida tomando en cuenta el tiempo limitado y enfocado a un solo autor.
- **Facilidad de Uso:** Se refiere a la dificultad que representa el usar la metodología y su puesta en práctica, se requiere indispensablemente este apartado por el tiempo limitado del proyecto, por lo tanto, se requiere dominar dicha metodología.
- **Documentación:** A pesar de que las metodologías ágiles no se enfoca en una documentación extensa, sino en los resultados, es necesario generar documentos según la metodología, se busca una metodología que no represente una extensa carga en la documentación.
- **Duración:** Se enfoca en el tiempo estimado que toma completar el proyecto aplicando cada metodología, al contar con un tiempo de desarrollo del proyecto limitado se requiere ganar tiempo aplicando una metodología que no requiera una duración extensa.

3.4.1. SCRUM

Se sugiere el uso de Scrum en proyectos que operan en entornos complejos, donde los requisitos experimentan cambios frecuentes y se requiere una entrega rápida de resultados. En estos casos, la innovación, la flexibilidad y la productividad son aspectos fundamentales a tener en cuenta (Ramírez et al., 2018)

Scrum es una metodología usada para gestionar proyectos que requieren una rápida adaptación al cambio. Permite dividir el proyecto en diferentes fases, lo que permite a los integrantes del equipo trabajar en roles específicos, fases y artefactos que permiten tener un mayor control, asegurando estándares de calidad y resultados satisfactorios.

Sprint: es una medida de tiempo utilizada en la metodología SCRUM, que dura entre una y cuatro semanas, donde el equipo de trabajo realizará la lista de tareas planificadas en un tiempo establecido.

3.4.1.1. Roles

En cuanto a los roles, SCRUM dispone de tres principales para la gestión y control del proyecto:

- **Product Owner:** El rol designado tiene la responsabilidad de establecer las características y requisitos del producto. Su función principal consiste en tomar decisiones sobre las funcionalidades clave del proyecto. De este modo, se asegura que el equipo de desarrollo tenga un entendimiento claro de los requerimientos del usuario.
- **Scrum Master:** El responsable de este rol tiene la tarea de asegurar el cumplimiento de las directrices y requisitos establecidos por el equipo de desarrollo. Actúa como guía en los diversos procesos, asegurando que el equipo cuente con las herramientas y recursos adecuados, cumpla con los plazos establecidos y realice entregas o avances que cumplan con los estándares de calidad.
- **Equipo de Desarrollo:** Es el grupo de desarrollo encargado de realizar las tareas y procesos designados según las funcionalidades. Disponen de conocimientos de análisis, diseño, programación y pruebas. Estos equipos de trabajo suelen ser multifuncionales.

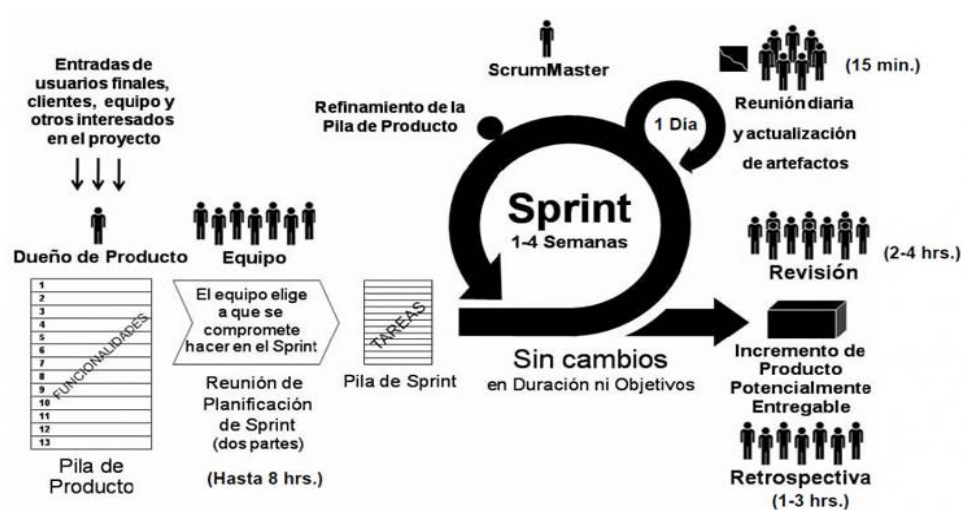
3.4.1.2. Artefactos

Los artefactos son los encargados de la documentación, permitiendo llevar un registro periódico de los avances, tareas y responsabilidades que el equipo de desarrollo mantiene. Estos son:

- **Product Backlog:** El Product Backlog lleva el registro de las funcionalidades, requerimientos del proyecto y determina las prioridades de cada fase del proyecto. El encargado de realizar actualizaciones y llevar un registro continuo es el Product Owner.
- **Sprint Backlog:** El Sprint Backlog es una lista de tareas que el equipo de desarrollo tiene que completar durante el periodo de tiempo o sprint previamente establecido.
- **Incremento:** Es el resultado obtenido en cada sprint. De esta forma, se puede revisar y mantener la calidad de los entregables, permitiendo tener un producto final de calidad.

Figura 1

Propuesta de método SIG con la metodología de desarrollo ágil



Nota: Adaptado de “Metodología Scrum en proyectos digitales - IDA Chile | Estrategia para el éxito de tu negocio” por Jordi Arino (2021)

3.4.1.3. Evaluación de los Criterios Establecidos

Se presenta a continuación el análisis realizado de la metodología SCRUM según los criterios definidos anteriormente:

- **Aprendizaje:** La curva de aprendizaje baja – media, ya que dispone de una estructura bien definida, permitiendo explicar y usar a detalle la metodología, además dispone de guías bien detalladas para cada paso o apartado de la metodología.
- **Facilidad de Uso:** SCRUM dispone de una Facilidad de Uso baja – media, ya que es una metodología ágil que se enfoca en la entrega continua y adaptación, aunque puede generar cierta dificultad si es que no se emplea adecuadamente la metodología, por esta razón se requiere que el equipo de trabajo tenga conocimiento sobre su uso.
- **Documentación:** SCRUM dispone de un amplio catálogo de recursos como guías, documentos, buenas prácticas, que sirven de ayuda para un correcto uso de la metodología.
- **Duración:** Tiene un promedio de 1 a 4 semanas cada Sprint.

3.4.2. Extreme Programming (XP)

La metodología Extreme Programming se enfoca en crear productos de software de manera rápida y eficiente, esta metodología busca cumplir las necesidades y requerimientos de los clientes en un periodo corto de tiempo. La metodología Extreme Programming se basa en: alcance, calidad, costo y tiempo.

3.4.2.1. Características

Según el autor Martínez (2019) la metodología Extreme Programming dispone de las siguientes características:

- **Desarrollo iterativo e incremental:** Requiere realizar pequeñas mejoras secuenciales de forma continua.
- **Pruebas unitarias continuas:** Se realizan pruebas frecuentes, repetidas y automatizadas, incluyendo pruebas de regresión que ayudarán a detectar errores en la funcionalidad.
- **Programación en grupo:** Es recomendable que dos a más personas desarrollen en conjunto ya que el código se revisará por más personas permitiendo asegurar la calidad del trabajo de desarrollo mejorando la productividad.
- **Corrección de errores:** Se prioriza la resolución de errores antes de añadir nuevas funcionalidades, esto aumenta el número de documentos entregables.
- **Integración del equipo de programación con el cliente/usuario:** Es recomendable que un representante o cliente trabaje en conjunto con el equipo de trabajo.
- **Refactorización del código:** En este apartado busca reescribir partes del código en base a características importantes como su legibilidad y mantenibilidad, sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- **Propiedad del código compartida:** Esta metodología promueve que todos los miembros del equipo pueden corregir y extender cualquier parte del proyecto, asegurando la colaboración y la responsabilidad. Las pruebas de regresión ayudan a detectar posibles errores.
- **Simplicidad en el código:** Se busca un desarrollo "simple" para que las cosas funcionen de forma eficiente, permitiendo realizar trabajo extra solo cuando se necesite una mejora, en lugar de utilizar un software y un código complejo que haga complicado cualquier cambio.

3.4.2.2. Roles.

Según el autor Martínez (2019) la metodología Extreme Programming tiene siete roles para los integrantes del proyecto:

- **Ciente / Usuario:** Este rol tiene la responsabilidad de redactar los requisitos del cliente y participar en las pruebas funcionales del producto para verificar su implementación. Además, se encarga de categorizar las tareas según su dificultad o importancia.
- **Coach:** Este rol tiene la responsabilidad de garantizar la correcta implementación de la metodología XP. Es necesario tener un conocimiento profundo de los procedimientos de XP para guiar al equipo durante todo el proceso de desarrollo del proyecto.
- **Big Boss:** Su principal objetivo es actuar como el enlace entre el cliente y el equipo de desarrollo. Su función principal consiste en llevar a cabo tareas de coordinación para asegurar una comunicación efectiva y fluida entre ambas partes.
- **Programador:** Los programadores desempeñan un papel fundamental como los principales creadores del producto. Es crucial que mantengan una comunicación constante y efectiva con los diversos miembros del equipo.
- **Tester:** Encargado de realizar las pruebas de funcionamiento del producto, mantiene una comunicación con el cliente y el equipo de trabajo.
- **Tracker:** Es el responsable de monitorear el avance del proyecto, comprobando que los plazos y entregas establecidos se cumplan. Debe realizar ajustes al plan en caso de imprevistos o modificaciones.

3.4.2.3. Evaluación de los Criterios Establecidos

Se presenta a continuación el análisis realizado de la metodología SCRUM según los criterios definidos anteriormente:

- **Aprendizaje:** La curva de aprendizaje media, por su enfoque en programación ágil y sus prácticas únicas, es muy similar a la metodología SCRUM.
- **Facilidad de Uso:** Tiene una facilidad de uso Media XP requiere que el equipo posea conocimiento sobre la metodología ya que requiere llevar correctamente los distintos procesos y la comunicación con el cliente.
- **Documentación:** XP dispone de un amplio catálogo de recursos como guías, documentos, buenas prácticas, que sirven de ayuda para un correcto uso de la metodología.
- **Duración:** Tiene un promedio de 2 a 3 semanas por tarea.

3.4.3. Comparativa de las Metodologías Ágiles

En base a los 4 criterios de evaluación de las metodologías ágiles mencionados, se definirá una tabla comparativa para concluir qué Metodologías Ágiles cumple con estos requerimientos para la elaboración del trabajo de titulación.

Tabla 4

Comparación de Metodologías Ágiles

| Puntos | XP | SCRUM |
|--------------------|--|---|
| Aprendizaje | La metodología XP dispone de una dificultad media por requerir contacto con el cliente | La Metodología SCRUM dispone de una curva de aprendizaje baja – media |

| | | |
|-------------------------|---|--|
| Facilidad de Uso | Requiere de alto conocimiento previo ambas metodologías | Se escogió SCRUM por su capacidad de adaptabilidad ante imprevistos o cambios. |
| Documentación | La metodología XP genera de 3 a 6 fases documentación. | Se escogió SCRUM por disponer de menor documentación, únicamente requiere documentación por cada sprint. |
| Duración | | Se escogió SCRUM por disponer de tiempos más cortos en la realización de tareas o sprints |
| Total | 4 | |

Autor: Cadena, S. (2023)

3.5. Herramientas Tecnológicas

Las herramientas Tecnológicas disponen de características propias y beneficios dependiendo el enfoque del proyecto, por esta razón, se detalla puntos importantes para definir que herramientas tecnológicas se adapta más a los propósitos del presente tema de titulación:

- **Costo:** Al ser un proyecto sin apoyo financiero se requiere Herramientas gratuitas o de bajo costo.
- **Aprendizaje:** Se requiere de herramientas intuitivas y sencillas que permitan solucionar requerimientos de forma rápida tomando en cuenta el tiempo limitado.
- **Documentación:** La herramienta tecnológica debe contar con basta documentación actualizada y soporte de la comunidad para brindar solución a requerimientos o inconvenientes de funcionamiento.

3.5.1. Entorno de desarrollo de código

La principal ventaja de utilizar un IDE es que proporciona una interfaz unificada y simplificada para el desarrollo de software, lo que ayuda a mejorar la productividad y la eficiencia del desarrollador (Red Hat, 2023). Al tener todas las herramientas necesarias integradas en un solo entorno, los desarrolladores pueden escribir, depurar y probar su código de manera más rápida y conveniente.

3.5.1.1. Visual Studio Code

Visual Studio Code es un editor de código fuente ligero pero potente que combina la simplicidad de un editor de texto con funciones avanzadas para la edición, depuración y administración de proyectos (Microsoft, 2023). Está diseñado para ser altamente personalizable, lo que permite a los desarrolladores adaptarlo a sus necesidades y preferencias. Con una interfaz de usuario intuitiva y una amplia gama de extensiones disponibles, VS Code ofrece una experiencia de desarrollo productiva y eficiente. Visual Studio Code (VS Code) es un entorno de desarrollo de código fuente abierto y gratuito desarrollado por Microsoft. Se ha convertido en una de las herramientas de desarrollo más populares utilizadas por programadores de diferentes lenguajes y plataformas.

3.5.1.2. Comparación de entornos de Desarrollo Web

Algunos ejemplos populares de IDE son Visual Studio (para el desarrollo de aplicaciones de Microsoft), Eclipse (para desarrollo en Java), IntelliJ IDEA (para desarrollo en Java, Kotlin y otros lenguajes), PyCharm (para desarrollo en Python) y Visual Studio Code, que, aunque se clasifica como un editor de código fuente, también proporciona muchas características de un IDE y se puede extender con numerosas extensiones.

Tabla 5

Comparación de entornos de Desarrollo Web

| | Visual Studio | IntelliJ IDEA | Visual Studio Code |
|-----------------------------------|--|---|---|
| Desarrollador principal | Microsoft | JetBrains | Microsoft |
| Enfoque principal | Desarrollo de aplicaciones de Microsoft, incluyendo aplicaciones de escritorio, web, móviles y de la nube (Microsoft, 2023). | Desarrollo de aplicaciones en Java y otros lenguajes relacionados (JetBrains, 2023). | Editor de código extensible con características de IDE (Microsoft, 2023). |
| Lenguajes admitidos | C#, VB.NET, F#, JavaScript, TypeScript y otros. | Java, Kotlin, Scala, Groovy, JavaScript, TypeScript y otros | Amplio soporte para múltiples lenguajes de programación, incluyendo C#, JavaScript, TypeScript, Python, Java, y muchos otros. |
| Características destacadas | Amplia gama de herramientas y plantillas específicas para desarrollar aplicaciones de Microsoft, depuración avanzada, integración con Azure, soporte de control de versiones y colaboración en equipo. | Potente editor de código con análisis estático avanzado, completado automático inteligente, refactorización de código, depuración avanzada, soporte para frameworks populares como Spring, Android y Maven, y una amplia gama de plugins. | Interfaz ligera pero potente, con una amplia gama de extensiones para personalizar y ampliar las capacidades del editor, soporte para depuración, control de versiones, herramientas de terminal integradas y una comunidad activa de desarrollo. |

Autor: Cadena, S. (2023)

En la tabla 4 se presenta un cuadro comparativo de los entornos de Desarrollo web en el cual se definen los puntos importantes para determinar qué entorno se adapta más a las necesidades del proyecto. Por lo tanto, podemos concluir que Visual Studio Code cumple con los requisitos y se adapta a las necesidades del trabajo de titulación.

3.5.2. Base de Datos

3.5.2.1. MySQL

MySQL es un sistema de gestión de base de datos relacional reconocida a nivel mundial por ser uno de los gestores de base de datos más usados de los últimos años junto con Oracle DB y Microsoft SQL Server. Permite almacenar y gestionar grandes cantidades de datos, siendo una opción ideal para el desarrollo de aplicaciones web y empresariales que requieran de escalabilidad y un alto rendimiento.

MySQL es una base de código abierto y gratuita, dispone de una amplia gama de herramientas que facilitan la administración y gestión de la base de datos, ofreciendo seguridad y confiabilidad sobre los datos.

3.5.2.2. Ventajas y Desventajas

Tabla 6

Tabla de Ventajas y Desventajas de la Base de Datos

| Ventajas | Desventajas |
|---|---|
| <ul style="list-style-type: none"> MySQL dispone de un interfaz sencillo, en conjunto de una instalación y configuración sencilla, permitiendo reducir tiempo en la comprensión del Gestor de Base de Datos. | <ul style="list-style-type: none"> MySQL al ser una herramienta gratuita no ofrece características de pago, tales como soporte y mayor grado de robustez. Al ser un software libre existe soluciones deficientes o limitadas, |

-
- MySQL es un gestor de Base de datos de código abierto y gratuito, cumpliendo los requerimientos del trabajo de titulación previamente mencionados.
 - MySQL permite el crecimiento del negocio al estar diseñado para manejar grandes volúmenes de datos, siendo ideal para la escalabilidad de pequeños y medianos negocios.
 - MySQL permite el procesamiento de grandes cantidades de datos de forma rápida, permitiendo tener respuestas y solicitudes de manera rápida y eficiente.
- que se convierten en una limitante cuando se busca el crecimiento y manejo de grandes empresas.
 - MySQL tiene una documentación limitada, aunque dispone de una amplia comunidad requiere de investigación para encontrar una solución acorde o temporal.
 - MySQL brinda una escalabilidad horizontal la cual requiere un alto grado de control y requerimientos de software y hardware.

Autor: Cadena, S. (2023)

3.5.2.3. Conclusión

MySQL nos permite dar el paso a la tecnología, siendo una herramienta recomendable para iniciar el desarrollo, manejo y administración de datos de un negocio, también en conjunto se contempla la migración de los datos desde la misma aplicación siendo beneficioso en caso de requerir cambio de gestor de base de datos a futuro.

3.5.3. Repositorio

3.5.3.1. GitHub

GitHub es una plataforma que permite el registro remoto del desarrollo de proyectos, facilitando la colaboración y el intercambio de proyectos entre usuarios. Además, ofrece funciones adicionales como la seguridad en la nube y otras características destacadas.(Astigarraga & Cruz-Alonso, 2022, p. 1)

GitHub es el repositorio más famoso de código en internet, permite llevar un control de versiones de forma remota, siendo beneficiosa para un grupo de trabajo, además, cuenta con una interfaz sencilla facilitando su manejo, cuenta con su versión de paga y gratuita, permitiendo brindar un servicio acorde a las necesidades del proyecto.

3.5.3.2. Ventajas y Desventajas

Tabla 7

Tabla de Ventajas y Desventajas del Repositorio

| Ventaja | Desventaja |
|--|---|
| <ul style="list-style-type: none"> • Github brinda control de versiones, permite a los desarrolladores trabajar sin afectar el código, esto gracias a que incluye seguimiento de los cambios realizados, y en caso de errores volver a una versión anterior. • GitHub cuenta con una vasta comunidad muy activa, permite realizar cambios, solventar problemas o errores presentados en menor tiempo. • GitHub permite el trabajo colaborativo, brindando flexibilidad al grupo de trabajo en cuestión de carga de trabajo, a su vez es compatible con una gran lista de lenguajes de programación y sistemas operativos. | <ul style="list-style-type: none"> • Existen funcionalidades que solo se incluyen en la versión de paga, permitiendo integrar herramientas adicionales o trabajar en proyectos privados. • En caso de que GitHub presente intermitencia o problemas con la plataforma puede significar una interrupción en el desarrollo. • El aprendizaje de GitHub es extenso, ya que existe una gran variedad de funcionalidades, herramientas y servicios que pueden ser complicados cuando se inicia. |

Autor: Cadena, S. (2023)

3.5.3.3. Conclusión

La plataforma GitHub con su versión gratuita es suficiente para el control de versiones de código, además de incluir beneficios como control de errores, trabajo colaborativo e integración con diferentes herramientas y una de las más importantes su comunidad y soporte, siendo un apoyo para el presente desarrollo del trabajo de titulación.

CAPÍTULO IV: DESARROLLO DE LA INVESTIGACIÓN

4. Aplicación de la Metodología SCRUM

En este capítulo se detalla los pasos necesarios para iniciar la metodología SCRUM, se detalla lo que se realizó, como se realizó el proyecto, también se define los encargados de los diferentes roles de la metodología SCRUM. Se detalla que es lo que se requiere implementar.

Se propone que el presente apartado determinar los diferentes requerimientos funcionales y no funcionales que ayuden a realizar los procesos y necesidades de un restaurante al implementar un aplicativo web que permita la gestión del control de reservas y administración del restaurante, tanto en la administración de inventario, permitiendo así un correcto control y organización del restaurante y una mejora en los tiempos de los procesos, tanto en el restaurante como del cliente.

4.1. Identificación de Roles

La organización del proyecto se basa en los principios de la metodología SCRUM, su enfoque es el progreso por etapas o Sprints. En cada etapa hay un responsable asignado a un rol específico, como se detalla a continuación:

Tabla 8

Roles y responsables

| Roles | Responsabilidades | Responsable |
|---------------|---|--------------------------------|
| Product Owner | <ul style="list-style-type: none">• Supervisar la implementación de la metodología Scrum, revisando el correcto uso y desarrollo de esta.• Coordinar los recursos del equipo, empleando herramientas que faciliten la gestión y desarrollo del proyecto. | Sebastián Andrés Cadena Molina |

| | | |
|---------------|--|--------------------------------|
| | <ul style="list-style-type: none"> • Detectar y solucionar los contratiempos que puedan surgir en el proceso. | |
| Scrum Master | <ul style="list-style-type: none"> • Establecer los requisitos funcionales del aplicativo web. • Representar a los clientes interesados en el aplicativo web. • Verificar el cumplimiento de los entregables de cada etapa en los plazos establecidos. • Aprobar o rechazar los entregables. | Sebastián Andrés Cadena Molina |
| Desarrollador | <ul style="list-style-type: none"> • Desarrollar el aplicativo web. • Realizar pruebas y entregar nuevas versiones del aplicativo web. | Sebastián Andrés Cadena Molina |

Autor: Cadena, S. (2023)

4.2. Identificación de Requerimientos

En este apartado, se busca mostrar los requerimientos funcionales y no funcionales que contribuyan a satisfacer las necesidades de un restaurante al implementar un sistema web que simplifique la gestión de los procesos de reservas, gestión de empleados, gestión de clientes y manejo de inventario. El objetivo de este sistema es mejorar el control y la organización del restaurante, reduciendo tiempo para el restaurante.

En la siguiente tabla se detalla las requisitos y procesos que se requieren automatizar y contemplar el sistema web:

4.2.1. Reunión Product Owner

Se requiere crear un sistema que permita la gestión del restaurante y automatizar sus distintos procesos.

Tabla 9

Identificación de requerimientos

| Número de Requerimiento | Requerimiento |
|--------------------------------|--|
| 1 | Se necesita un apartado que pueda evidenciar los diferentes platos que ofrece el restaurante |
| 2 | Se necesita tener un apartado donde contenga la información de los pedidos del restaurante |
| 3 | Desarrollar un sistema donde el usuario pueda realizar reservas en el local |
| 4 | Tener un apartado donde contenga la información de los productos del inventario |
| 5 | Tener un apartado donde contenga información importante del restaurante donde pueda promocionarse. |
| 6 | Tener un apartado donde contenga la información de los empleados del restaurante |
| 7 | Tener un apartado con la información de los clientes del restaurante |
| 8 | Desarrollar una base de datos para almacenar los datos necesarios para la gestión del restaurante |

Autor: Cadena, S. (2023)

4.2.2. Requerimientos Funcionales

La característica principal del sistema web que se va a desarrollar es proporcionar un mayor control sobre las reservas y la administración del restaurante a través de un sistema web, así como también brindar un mayor control sobre los diferentes procesos y requisitos para el funcionamiento óptimo del restaurante.

El sistema que se va a desarrollar va a ser utilizado tanto por el administrador de la página web, como por los clientes del restaurante, por lo que los requerimientos

funcionales que se van a tratar están enfocados para el uso de los distintos usuarios sin que la página se sature.

4.2.2.1. Administrador del Restaurante

En la siguiente tabla de requerimientos funcionales se detalla lo que va a poder realizar el sistema web como administrador:

Tabla 10

Priorización de requerimientos

| Requerimientos Funcionales | | Prioridad | Descripción |
|----------------------------|--|-----------|--|
| RF1. | Gestionar Plato | Alta | Se requiere ingresar nuevos Platos, modificar, eliminar y la visualización de Platos creados por el encargado de Cocina. |
| RF2. | Gestionar Pedidos | Media | Se requiere realizar pedidos, modificarlos, eliminarlos y visualizar los pedidos creados por el cliente. |
| RF3. | Gestionar Reservas | Alta | Se requiere realizar nuevas reservas, modificarlas, eliminarlas y visualizar las reservas creados por los clientes. |
| RF4. | Gestionar Producto | Alta | Se requiere un apartado donde se pueda gestionar el ingreso de nuevos Productos, modificarlos, eliminarlos y la visualización de productos en el Inventario. |
| RF5. | Ver información de productos del Restaurante | Media | Se requiere un apartado para ver la información que, del |

| | | | |
|------|---------------------|------|--|
| | | | restaurante, sus Platos, reservas y datos sobre el mismo. |
| RF6. | Gestionar Empleados | Alta | Se requiere almacenar los datos de los empleados del restaurante |
| RF7. | Gestionar Clientes | Alta | Se requiere almacenar los datos de los clientes del restaurante |

Autor: Cadena, S. (2023)

4.2.2.2. Cliente del Restaurante

En la siguiente tabla de requerimientos funcionales se detalla lo que va a poder realizar el sistema web como Cliente:

Tabla 11

Requerimientos funcionales

| Requerimientos Funcionales | | Prioridad | Descripción |
|----------------------------|--|-----------|--|
| RF1. | Gestionar Pedidos | Media | Se requiere un apartado para realizar pedidos, y debe visualizar los pedidos creados. |
| RF2. | Gestionar Reservas | Alta | Se requiere un apartado para realizar nuevas reservas. |
| RF3. | Ver información de productos del Restaurante | Media | Se requiere un apartado donde se pueda conocer sobre los productos, Platos, y servicios en el restaurante. |

Autor: Cadena, S. (2023)

4.2.3. Requerimientos no funcionales

Se requiere que el sistema web cuente con un interfaz amigable para el usuario, enfocado en brindar al usuario una correcta orientación y un manejo intuitivo permitiendo al usuario acoplarse a la brevedad al sistema.

La siguiente tabla muestra los Requerimientos no Funcionales para el aplicativo web.

Tabla 12

Requerimientos no Funcionales

| Requerimientos No Funcionales | Prioridad |
|---|------------------|
| Usabilidad – El aplicativo web debe trabajar sobre los navegadores web más populares, tanto para computadoras de escritorio como para dispositivos móviles, incluyen Google Chrome, Mozilla Firefox y Microsoft Edge. Estas aplicaciones permiten a los usuarios acceder y navegar por sitios web de manera eficiente y brindan diversas funciones y características para mejorar la experiencia de navegación. | Alta |
| Responsivo – El sistema web debe visualizarse correctamente en diferentes dispositivos (computadoras, tabletas, celulares) | Alta |
| Disponibilidad – El sistema web debe estar disponible el 99.99% de las veces en que el usuario intente ingresar. | Alta |
| El sistema web debe mostrar mensajes útiles que permitan orientar al usuario final | Media |

Autor: Cadena, S. (2023)

4.2.4. Product Backlog

El Product Backlog lleva el registro de las funcionalidades, requerimientos del proyecto y determina las prioridades de cada fase del proyecto. El documento puede ser modificado en cualquier momento por cualquier integrante del grupo de trabajo.

Tabla 13

Product Backlog

| Priorización | Nombre | Descripción | Estimación |
|---------------------|---------------|--|-------------------|
| 1 | SP N°10 | Establecer Conexión con la Base de Datos | 7 |
| 2 | SP N°7 | Página CRUD de Empleados | 7 |
| 3 | SP N°8 | Página CRUD de Clientes | 7 |
| 4 | SP N°1 | Página CRUD de Plato | 10 |
| 5 | SP N°5 | Página CRUD de Productos | 7 |
| 6 | SP N°3 | Página CRUD de Reservas | 9 |
| 7 | SP N°2 | Página CRUD de Pedidos | 12 |

Autor: Cadena, S. (2023)

4.2.5. Sprint Planning Meeting

En esta sección el Product Owner es el encargado de priorizar las tareas del proyecto, donde se establecen tiempos de estimación para terminarlo.

Tabla 14

Sprint Planning Meeting – Sprint 1

| Priorización | Nombre | Descripción | Estimación |
|---------------------|---------------|--|-------------------|
| 1 | SP N°10 | Establecer Conexión con la Base de Datos | 7 |
| 2 | SP N°7 | Página CRUD de Empleados | 7 |
| 3 | SP N°8 | Página CRUD de Clientes | 7 |

Autor: Cadena, S. (2023)

Tabla 15

Sprint Planning Meeting – Sprint 2

| Priorización | Nombre | Descripción | Estimación |
|---------------------|---------------|-------------------------|-------------------|
| 4 | SP N°1 | Página CRUD de Plato | 10 |
| 5 | SP N°5 | Página CRUD de Producto | 7 |

Autor: Cadena, S. (2023)

Tabla 16

Sprint Planning Meeting – Sprint 3

| Priorización | Nombre | Descripción | Estimación |
|---------------------|---------------|-------------------------|-------------------|
| 7 | SP N°3 | Página CRUD de Reservas | 9 |
| 8 | SP N°2 | Página CRUD de Pedidos | 12 |

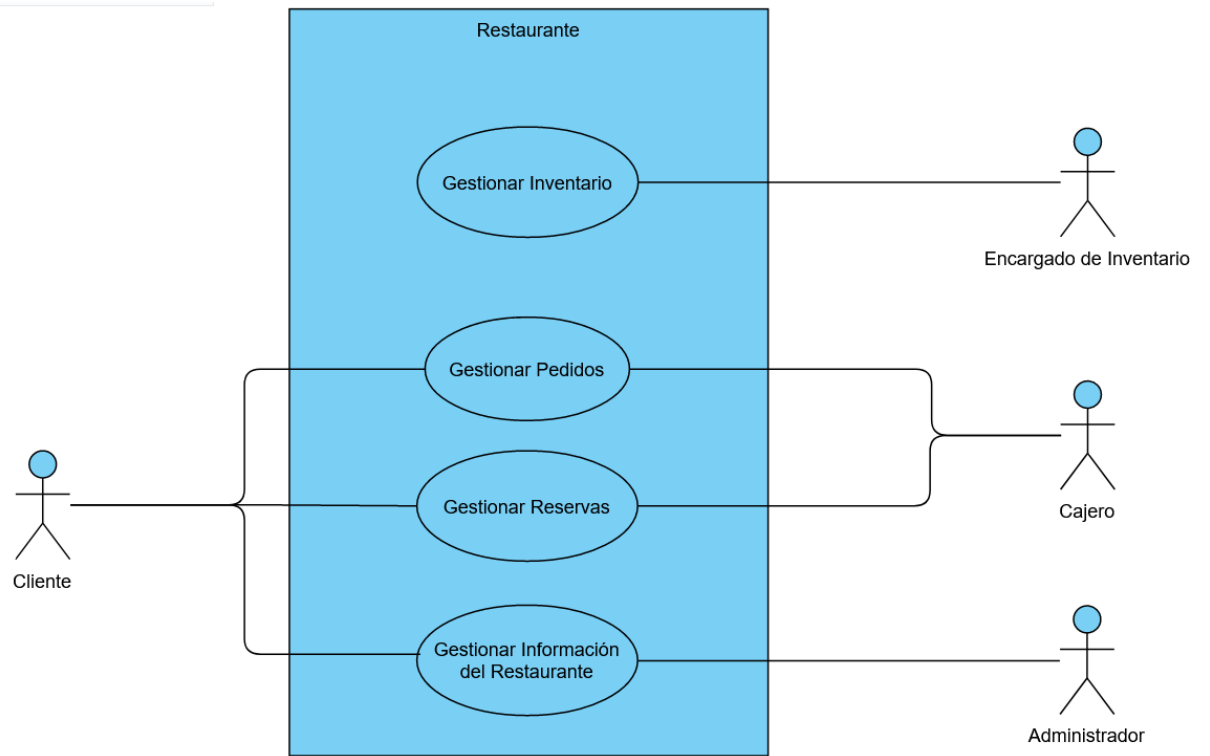
Autor: Cadena, S. (2023)

4.3. Diseño Lógico y Funcional

4.3.1. Diagrama General de Casos de Uso

Figura 2

Diagrama General de Casos de Uso



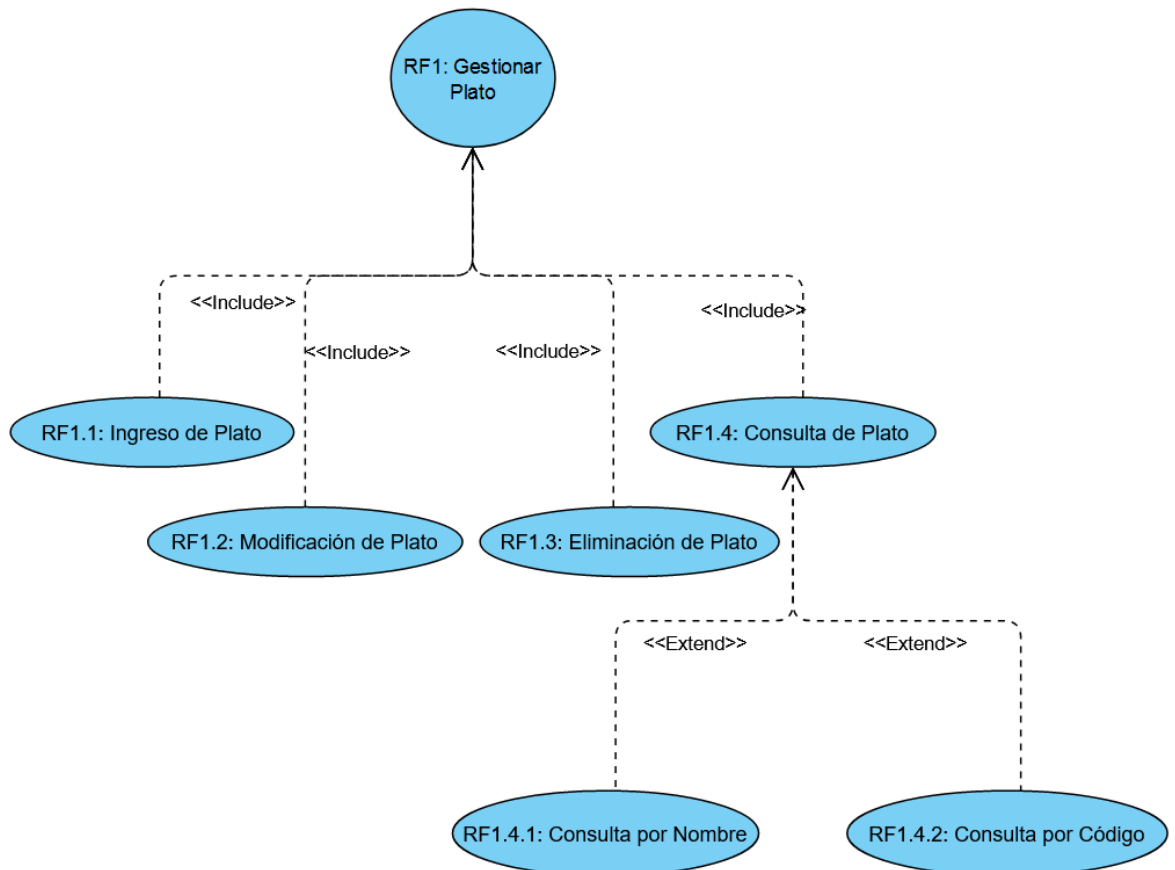
Autor: Cadena, S. (2023)

4.3.2. Nivel de Detalle de Casos de Uso

4.3.2.1. RF1 Gestionar Plato

Figura 3

Diagrama de Caso de Uso Siguiete Nivel - Gestionar Plato

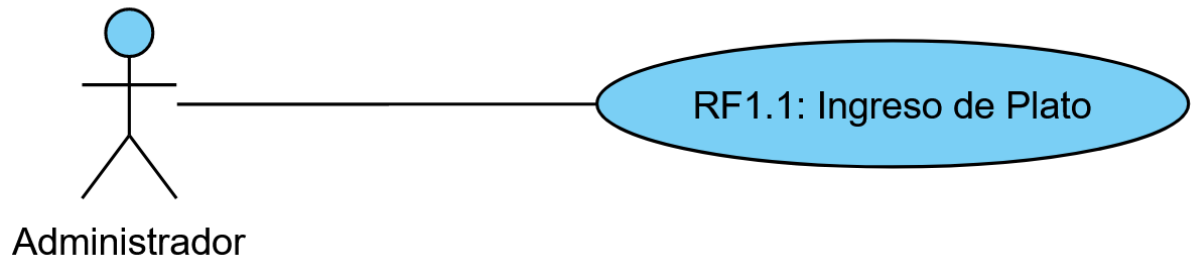


Autor: Cadena, S. (2023)

ID: RF1.1 Ingreso de Plato

Figura 4

Diagrama de Caso de Uso A Detalle - Ingreso de Plato



Autor: Cadena, S. (2023)

Actores: Administrador

Flujo Principal

1. El actor presiona la opción de Gestionar Plato.
2. El sistema despliega formulario de Gestión de Plato.
3. El actor selecciona la opción Ingreso de Plato.
4. El sistema despliega pantalla para Ingresar Plato.
5. El actor ingresa los datos necesarios de un nuevo Plato.
6. El actor presiona Guardar.
7. El sistema verifica su existencia (E1).
8. El sistema almacena la información (E2).

Flujo Alternativo

1. Puede seleccionar RF1.2 Modificación de Plato, RF1.3 Eliminación de Plato o RF1.4 Consulta de Plato.

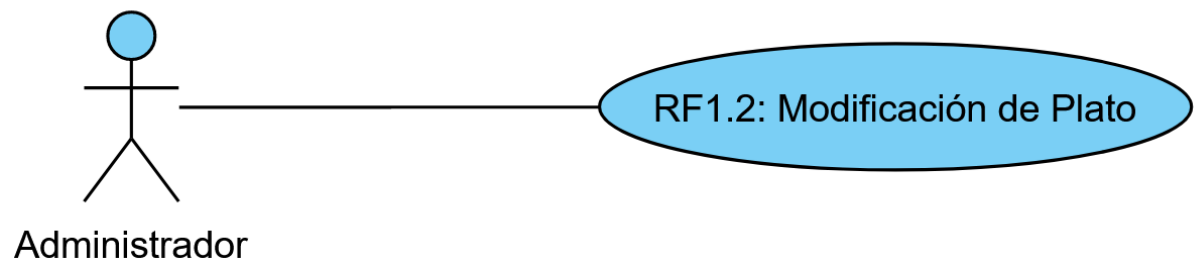
Excepciones

- E1 Si el Plato ya existe el sistema presentará un mensaje el cual indique que el Plato ya existe.
- E2 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF1.2 Modificación de Plato

Figura 5

Diagrama de Caso de Uso A Detalle - Modificación de Plato



Autor: Cadena, S. (2023)

Actores: Administrador

Flujo Principal

1. El actor presiona la opción de Gestionar Plato.
2. El sistema despliega formulario de Gestión de Plato.
3. El actor selecciona la opción Modificación de Plato.
4. El sistema despliega pantalla para modificar Plato.
5. El actor ingresa los datos necesarios para modificar el Plato.
6. El actor presiona Guardar.
7. El sistema Verifica la Información. (E1)
8. El sistema almacena la información. (E2)

Flujo Alternativo

1. Puede seleccionar RF1.1 Ingreso de Plato, RF1.3 Eliminación de Plato o RF1.4 Consulta de Plato

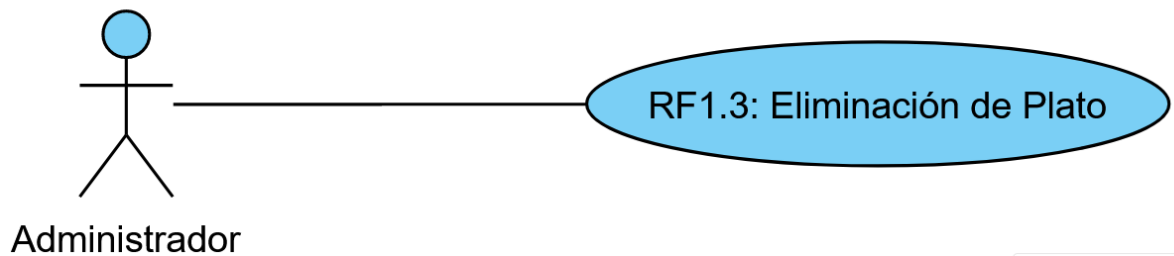
Excepciones

- E1 Si los datos del Plato están incompletos.
- E2 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF1.3 Dar de baja del Plato

Figura 6

Diagrama de Caso de Uso A Detalle – Eliminación de Plato



Autor: Cadena, S. (2023)

Actores: Administrador

Flujo Principal

1. El actor presiona la opción de Gestionar Plato.
2. El sistema despliega formulario de Gestión de Plato.
3. El actor selecciona la opción Eliminación de Plato.

4. El sistema despliega pantalla para Eliminación Plato.
5. El actor ingresa los datos necesarios.
6. El actor presiona Eliminar Plato.
7. El sistema desplegará un mensaje: "¿Está seguro de Eliminar el Plato?".
8. El usuario Selecciona Si
9. El sistema Eliminará la información (E1)

Flujo Alternativo

1. Puede seleccionar RF1.1 Ingreso de Plato, RF1.2 Modificación de Plato o RF1.4 Consulta de Plato
8. El usuario selecciona No, el sistema no Eliminará la información

Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF1.4.1 Consulta de Plato por Nombre

Figura 7

Diagrama de Caso de Uso A Detalle - Consulta de Plato por Nombre



Autor: Cadena, S. (2023)

Actores: Administrador

Flujo Principal

1. El actor presiona la opción de Gestionar Plato.
2. El sistema despliega formulario de Gestión de Plato.
3. El actor selecciona la opción Consulta de Plato.
4. El sistema despliega pantalla para Consulta de Plato.
5. El actor selecciona la opción Consulta de Plato por Código.
6. El sistema presenta la pantalla para Consulta de Plato por Código.
7. El actor ingresa el nombre que desea consultar.
8. El sistema verifica el Código.
9. El sistema desplegará información del Plato Consultado (E1)

Flujo Alternativo

1. Puede seleccionar RF1.1 Ingreso de Plato, RF1.2 Modificación de Plato o RF1.3 Eliminación de Plato

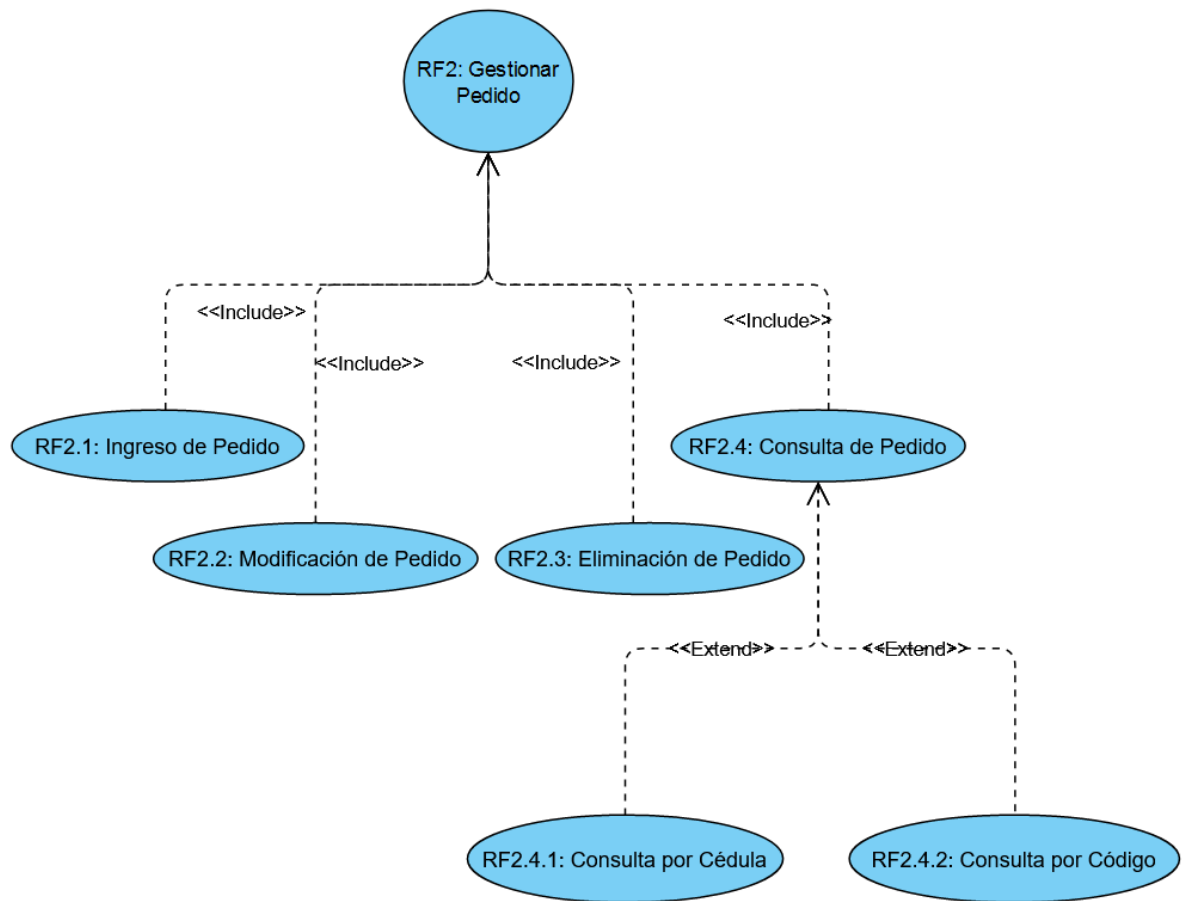
Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

4.3.2.2. RF2 Gestionar Pedido

Figura 8

Diagrama de Caso de Uso Siguiete Nivel - Gestionar Pedido

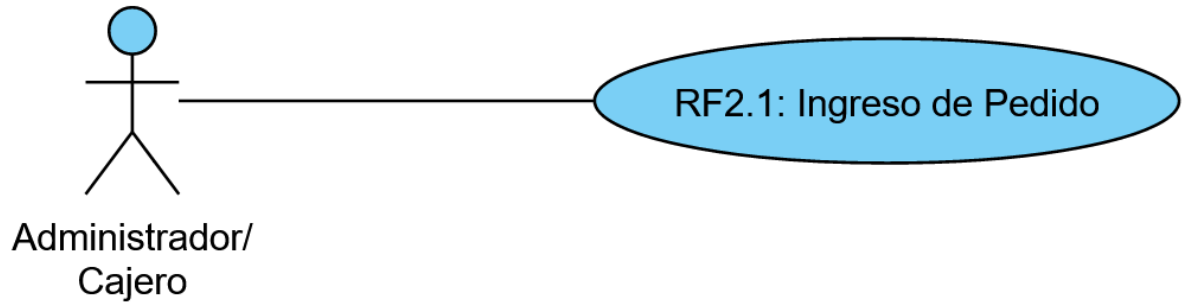


Autor: Cadena, S. (2023)

ID: RF2.1 Ingreso de Pedidos

Figura 9

Diagrama de Caso de Uso A Detalle - Ingreso de Pedido



Autor: Cadena, S. (2023)

Actores: Administrador, Cajero

Flujo Principal

1. El actor presiona la opción de Gestionar Pedidos.
2. El sistema despliega formulario de Gestión de Pedidos.
3. El actor selecciona la opción Ingreso de Pedidos.
4. El sistema despliega pantalla para Ingresar Pedidos.
5. El actor ingresa los datos necesarios de un nuevo Pedidos.
6. El actor presiona Guardar.
7. El sistema verifica la información (E1).
8. El sistema almacena la información (E2).

Flujo Alternativo

1. Puede seleccionar RF2.2 Modificación de Pedido, RF2.3 Eliminación de Pedido o RF2.4 Consulta de Pedido.

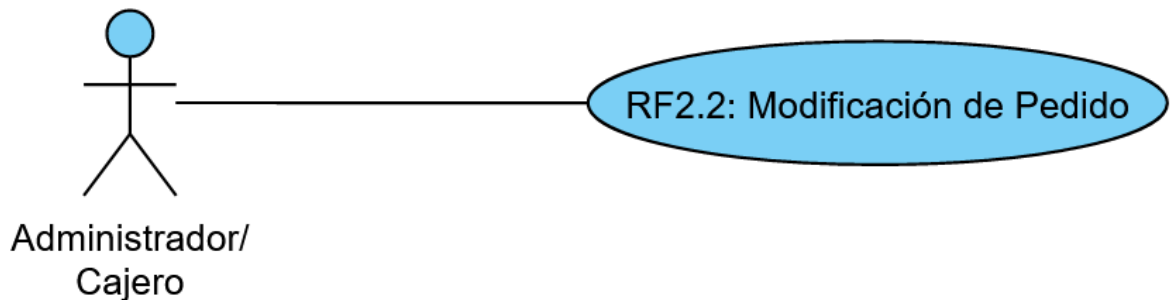
Excepciones

- E1 Si el Pedido tiene campos incompletos el sistema presentará un mensaje: "Complete los campos requeridos".
- E2 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF2.2 Modificación de Pedido

Figura 10

Diagrama de Caso de Uso A Detalle - Modificación de Pedido



Autor: Cadena, S. (2023)

Actores: Administrador, Cajero

Flujo Principal

1. El actor presiona la opción de Gestionar Pedido.
2. El sistema despliega formulario de Gestión de Pedido.
3. El actor selecciona la opción Modificación de Pedido.

4. El sistema despliega pantalla para modificar Pedido.
5. El actor ingresa los datos necesarios para modificar el Pedido.
6. El actor presiona Guardar.
7. El sistema Verifica la Información. (E1)
8. El sistema almacena la información. (E2)

Flujo Alternativo

1. Puede seleccionar RF2.1 Ingreso de Pedido, RF2.3 Eliminación de Pedido o RF2.4 Consulta de Pedido

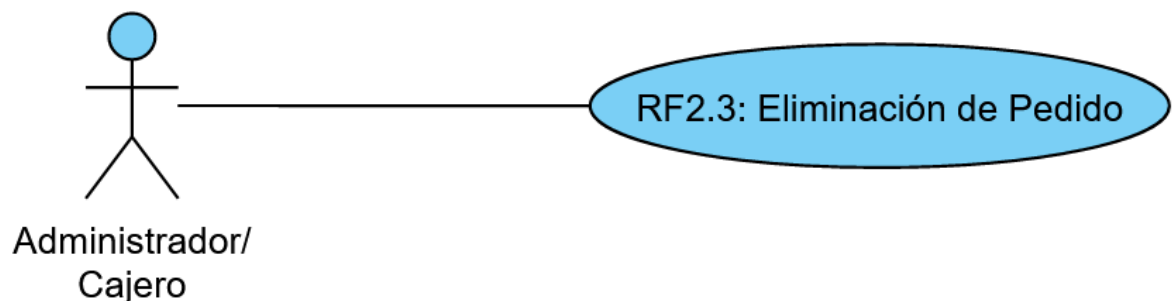
Excepciones

- E1 Si los datos del Pedido están incompletos.
- E2 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF2.3 Eliminación de Pedido

Figura 11

Diagrama de Caso de Uso A Detalle - Eliminación de Pedido



Autor: Cadena, S. (2023)

Actores: Administrador, Cajero

Flujo Principal

1. El actor presiona la opción de Gestionar Pedido.
2. El sistema despliega formulario de Gestión de Pedido.
3. El actor selecciona la opción Eliminación de Pedido.
4. El sistema despliega pantalla para Eliminación Pedido.
5. El actor ingresa los datos necesarios.
6. El actor presiona Eliminar Pedido.
7. El sistema desplegará un mensaje: "¿Está seguro de Eliminar el Pedido?".
8. El usuario Selecciona Si
9. El sistema Eliminará la información (E1)

Flujo Alterno

1. Puede seleccionar RF2.1 Ingreso de Pedido, RF2.2 Modificación de Pedido o RF2.4 Consulta de Pedido.
8. El usuario selecciona No, el sistema no Eliminará la información.

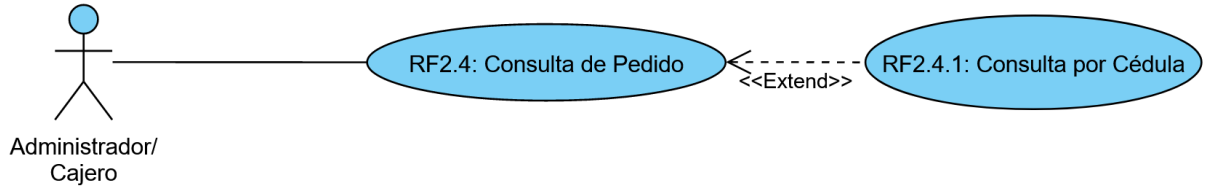
Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF2.4.1 Consulta de Pedidos por Cédula

Figura 12

Diagrama de Caso de Uso A Detalle - Consulta de Pedidos por Cédula



Autor: Cadena, S. (2023)

Actores: Administrador, Cajero

Flujo Principal

1. El actor presiona la opción de Gestionar Pedido.
2. El sistema despliega formulario de Gestión de Pedido.
3. El actor selecciona la opción Consulta de Pedido.
4. El sistema despliega pantalla para Consulta de Pedido.
5. El actor selecciona la opción Consulta de Pedido por Cédula.
6. El sistema presenta la pantalla para Consulta de Pedido por Cédula.
7. El actor ingresa la Cédula que desea consultar.
8. El sistema verifica la Cédula.
9. El sistema desplegará información del Pedido Consultado (E1).

Flujo Alternativo

1. Puede seleccionar RF2.1 Ingreso de Pedido, RF2.2 Modificación de Pedido o RF2.3 Eliminación de Pedido.
2. El usuario selecciona Consulta de Plato por Nombre.

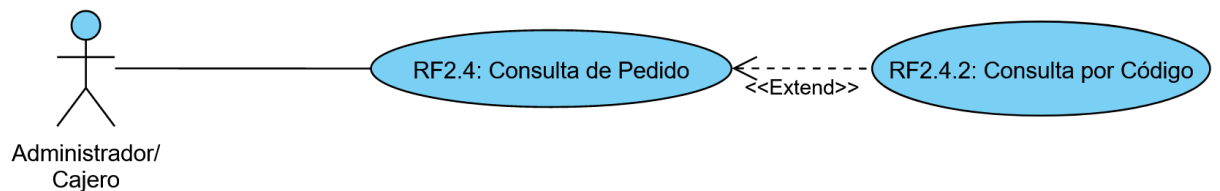
Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF2.4.2 Consulta de Pedidos por Código

Figura 13

Diagrama de Caso de Uso A Detalle - Consulta de Pedidos por Código



Actores: Administrador, Cajero

Flujo Principal

10. El actor presiona la opción de Gestionar Pedido.
11. El sistema despliega formulario de Gestión de Pedido.
12. El actor selecciona la opción Consulta de Pedido.
13. El sistema despliega pantalla para Consulta de Pedido.
14. El actor selecciona la opción Consulta de Pedido por Código.
15. El sistema presenta la pantalla para Consulta de Pedido por Código.
16. El actor ingresa la Código que desea consultar.
17. El sistema verifica la Código.
18. El sistema desplegará información del Pedido Consultado (E1).

Flujo Alternativo

3. Puede seleccionar RF2.1 Ingreso de Pedido, RF2.2 Modificación de Pedido o RF2.3 Eliminación de Pedido.
4. El usuario selecciona Consulta de Plato por Nombre.

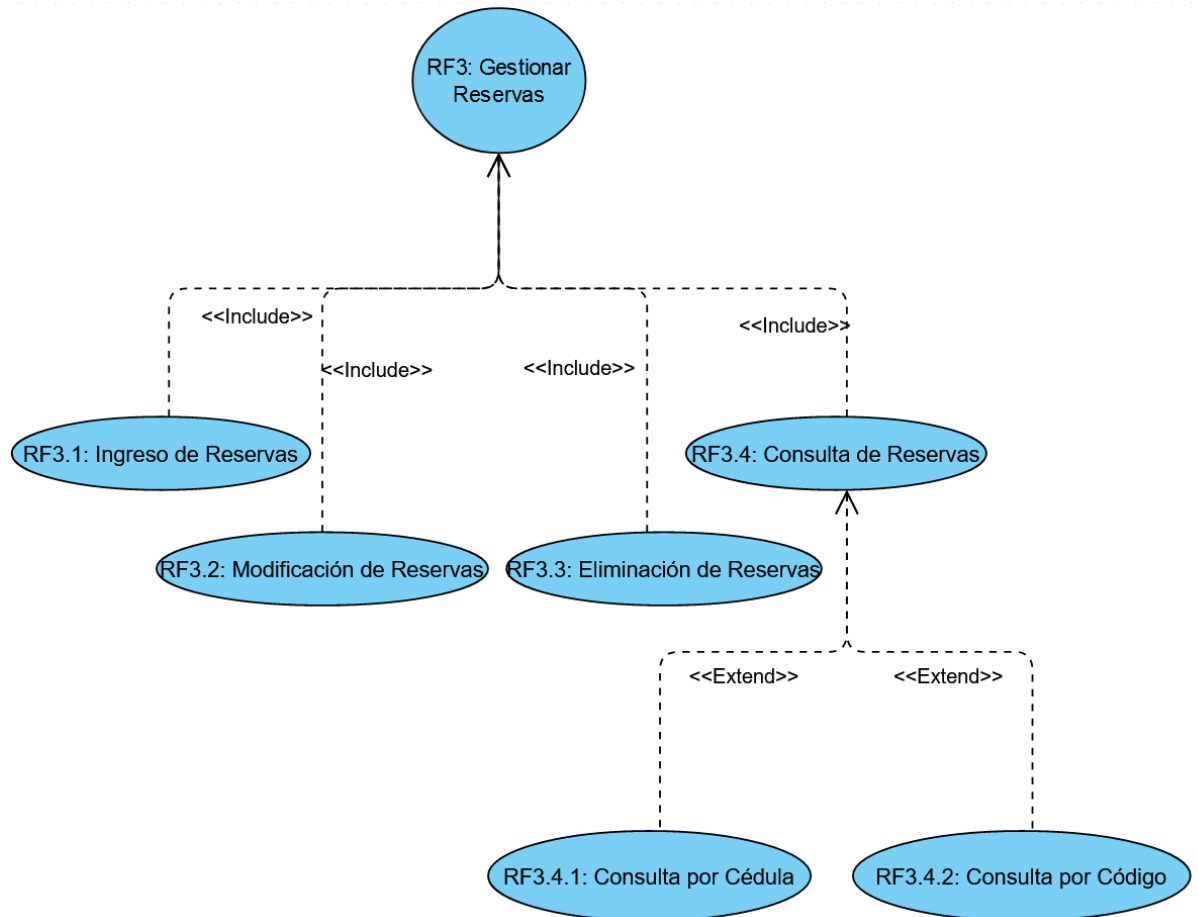
Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

4.3.2.3. RF3 Gestionar Reservas

Figura 14

Diagrama de Caso de Uso Siguiendo Nivel - Gestionar Reservas

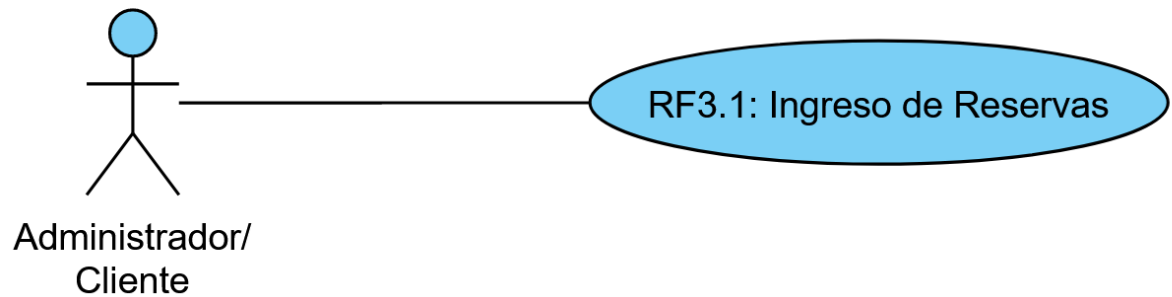


Autor: Cadena, S. (2023)

ID: RF3.1 Ingreso de Reservas

Figura 15

Diagrama de Caso de Uso A Detalle - Ingreso de Reservas



Autor: Cadena, S. (2023)

Actores: Administrador, Cliente

Flujo Principal

1. El actor presiona la opción de Gestionar Reserva.
2. El sistema despliega formulario de Gestión de Reserva.
3. El actor selecciona la opción Ingreso de Reserva.
4. El sistema despliega pantalla para Ingresar Reserva.
5. El actor ingresa los datos necesarios de un nuevo Reserva.
6. El actor presiona Guardar.
7. El sistema verifica su existencia (E1).
8. El sistema almacena la información (E2).

Flujo Alternativo

1. Puede seleccionar RF3.2 Modificación de Reserva, RF3.3 Eliminación de Reserva o RF3.4 Consulta de Reserva.

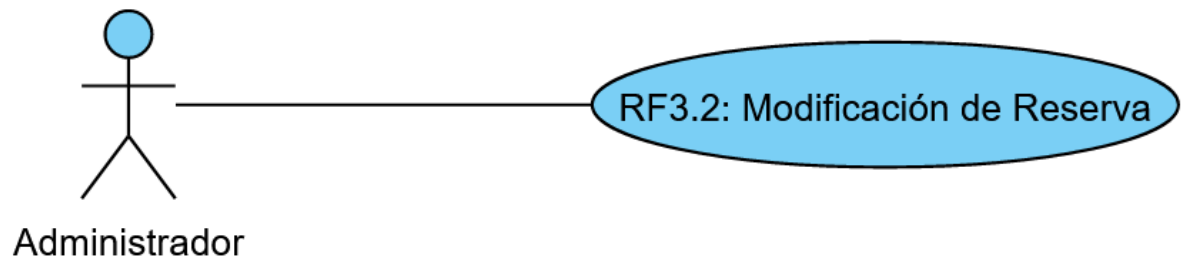
Excepciones

- E1 Si la Reserva ya existe el sistema presentará un mensaje el cual indique que la Reserva ya existe.
- E2 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF3.2 Modificación de Reserva

Figura 16

Diagrama de Caso de Uso A Detalle - Modificación de Reserva



Autor: Cadena, S. (2023)

Actores: Administrador

Flujo Principal

1. El actor presiona la opción de Gestionar Reserva.
2. El sistema despliega formulario de Gestión de Reserva.
3. El actor selecciona la opción Modificación de Reserva.
4. El sistema despliega pantalla para modificar Reserva.
5. El actor ingresa los datos necesarios para modificar el Reserva.
6. El actor presiona Guardar.

7. El sistema Verifica la Información. (E1)
8. El sistema almacena la información. (E2)

Flujo Alternativo

1. Puede seleccionar RF3.1 Ingreso de Reserva, RF3.3 Eliminación de Reserva o RF3.4 Consulta de Reserva.

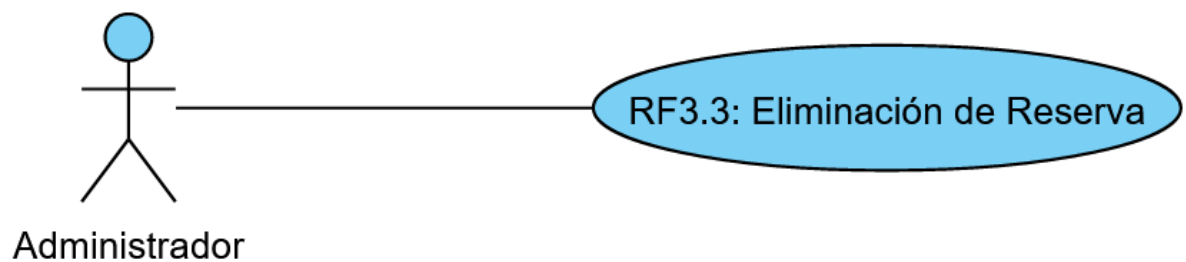
Excepciones

- E1 Si los datos del Reserva están incompletos.
- E2 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF3.3 Eliminación de Reserva

Figura 17

Diagrama de Caso de Uso A Detalle - Eliminación de Reserva



Autor: Cadena, S. (2023)

Actores: Administrador

Flujo Principal

1. El actor presiona la opción de Gestionar Reserva.

2. El sistema despliega formulario de Gestión de Reserva.
3. El actor selecciona la opción Eliminación de Reserva.
4. El sistema despliega pantalla para Eliminación Reserva.
5. El actor ingresa los datos necesarios.
6. El actor presiona Eliminar Reserva.
7. El sistema desplegará un mensaje: "¿Está seguro de Eliminar el Reserva?".
8. El usuario Selecciona Si
9. El sistema Eliminará la información (E1)

Flujo Alternativo

1. Puede seleccionar RF1.1 Ingreso de Reserva, RF1.2 Modificación de Reserva o RF1.4 Consulta de Reserva
8. El usuario selecciona No, el sistema no Eliminará la información

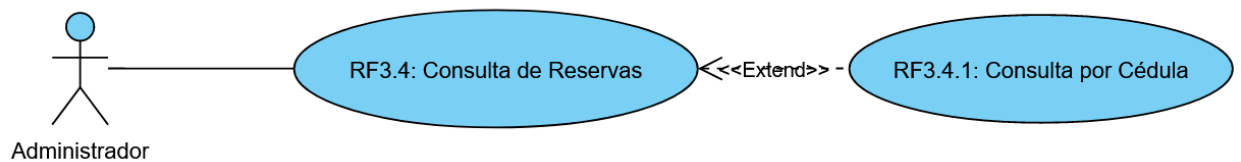
Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF3.4.1 Consulta de Reservas por Cédula

Figura 18

Diagrama de Caso de Uso A Detalle - Consulta de Reservas por Cédula



Autor: Cadena, S. (2023)

Actores: Administrador

Flujo Principal

1. El actor presiona la opción de Gestionar Reserva.
2. El sistema despliega formulario de Gestión de Reserva.
3. El actor selecciona la opción Consulta de Reserva.
4. El sistema despliega pantalla para Consulta de Reserva.
5. El actor selecciona la opción Consulta de Reserva por Cédula.
6. El sistema presenta la pantalla para Consulta de Reserva por Cédula.
7. El actor ingresa la Cédula que desea consultar.
8. El sistema verifica la Cédula.
9. El sistema desplegará información del Reserva Consultado (E1)

Flujo Alternativo

2. Puede seleccionar RF3.1 Ingreso de Reserva, RF3.2 Modificación de Reserva o RF3.3 Eliminación de Reserva
5. El usuario selecciona Consulta de Reserva por Nombre.

Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF3.4.2 Consulta de Reservas por Nombre

Figura 19

Diagrama de Caso de Uso A Detalle - Consulta de Reservas por Nombre



Autor: Cadena, S. (2023)

Actores: Administrador

Flujo Principal

1. El actor presiona la opción de Gestionar Reserva.
2. El sistema despliega formulario de Gestión de Reserva.
3. El actor selecciona la opción Consulta de Reserva.
4. El sistema despliega pantalla para Consulta de Reserva.
5. El actor selecciona la opción Consulta de Reserva por Código.
6. El sistema presenta la pantalla para Consulta de Reserva por Código.
7. El actor ingresa el Código que desea consultar.
8. El sistema verifica la Código.
9. El sistema desplegará información del Reserva Consultado (E1)

Flujo Alterno

1. Puede seleccionar RF3.1 Ingreso de Reserva, RF3.2 Modificación de Reserva o RF3.3 Eliminación de Reserva
5. El usuario selecciona Consulta de Reserva por Cédula.

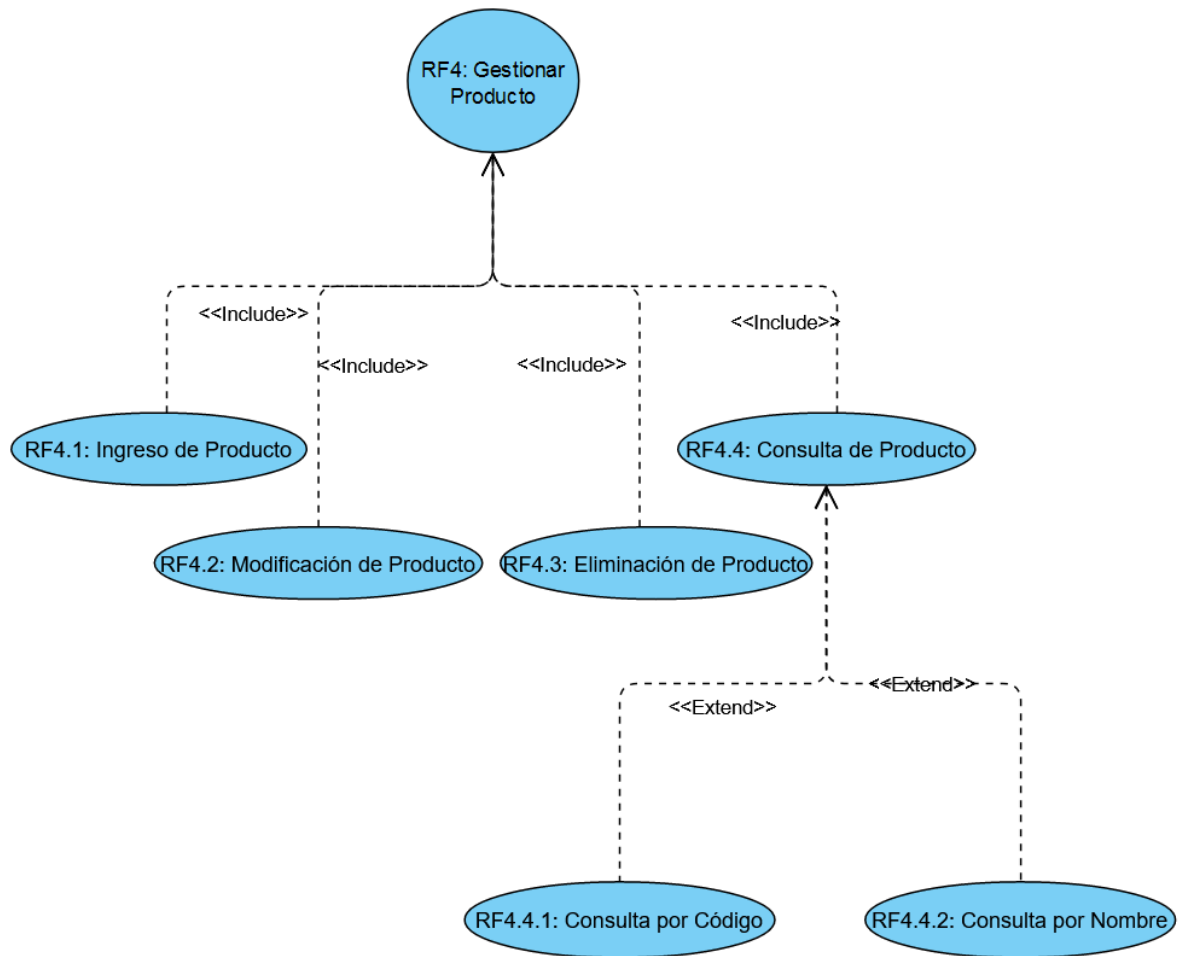
Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

4.3.2.4. RF4 Gestionar Producto

Figura 20

Diagrama de Caso de Uso Siguiendo Nivel - Gestionar Producto

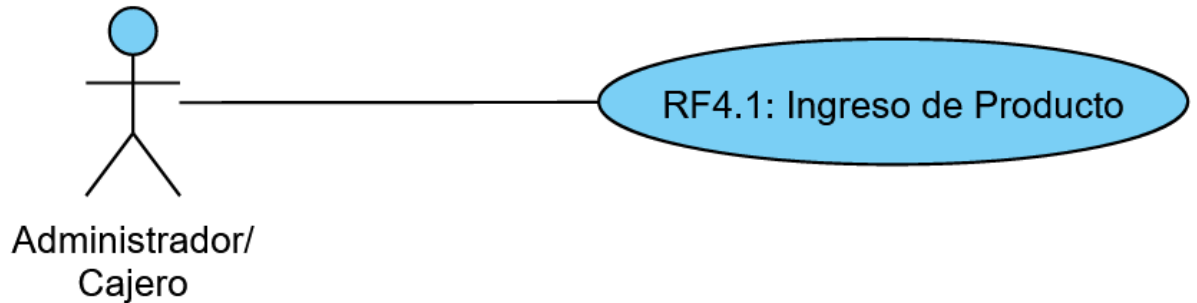


Autor: Cadena, S. (2023)

ID: RF4.1 Ingreso de Producto

Figura 21

Diagrama de Caso de Uso A Detalle - Ingreso de Producto



Autor: Cadena, S. (2023)

Actores: Administrador, Encargado Producto

Flujo Principal

1. El actor presiona la opción de Gestionar Producto.
2. El sistema despliega formulario de Gestión de Producto.
3. El actor selecciona la opción Ingreso de Producto.
4. El sistema despliega pantalla para Ingresar Producto.
5. El actor ingresa los datos necesarios de un nuevo Producto.
6. El actor presiona Guardar.
7. El sistema verifica su existencia (E1).
8. El sistema almacena la información (E2).

Flujo Alternativo

2. Puede seleccionar RF4.2 Modificación de Producto, RF4.3 Eliminación de Producto o RF4.4 Consulta de Producto.

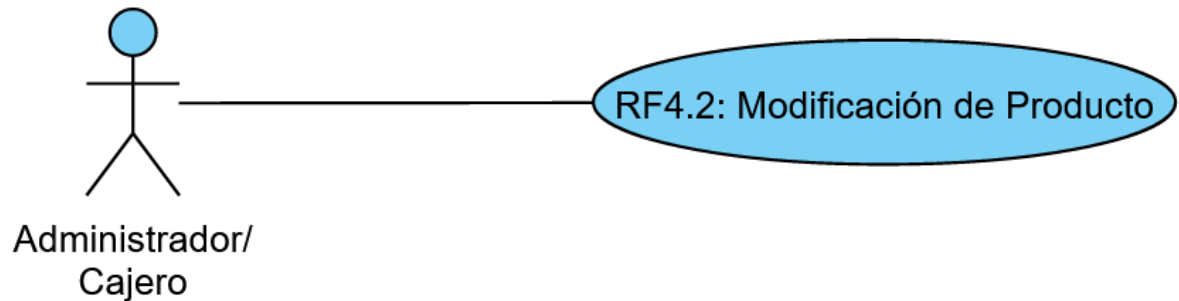
Excepciones

- E1 Si el producto de Producto ya existe el sistema presentará un mensaje el cual indique que el Producto ya existe.
- E2 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF4.2 Modificación de Producto

Figura 22

Diagrama de Caso de Uso A Detalle - Modificación de Producto



Autor: Cadena, S. (2023)

Actores: Administrador, Encargado Producto

Flujo Principal

9. El actor presiona la opción de Gestionar Producto.
10. El sistema despliega formulario de Gestión de Producto.
11. El actor selecciona la opción Modificación de Producto.
12. El sistema despliega pantalla para modificar Producto.
13. El actor ingresa los datos necesarios para modificar el Producto.
14. El actor presiona Guardar.
15. El sistema Verifica la Información. (E1)

16. El sistema almacena la información. (E2)

Flujo Alternativo

1. Puede seleccionar RF4.1 Modificación de Producto, RF4.3 Eliminación de Producto o RF4.4 Consulta de Producto

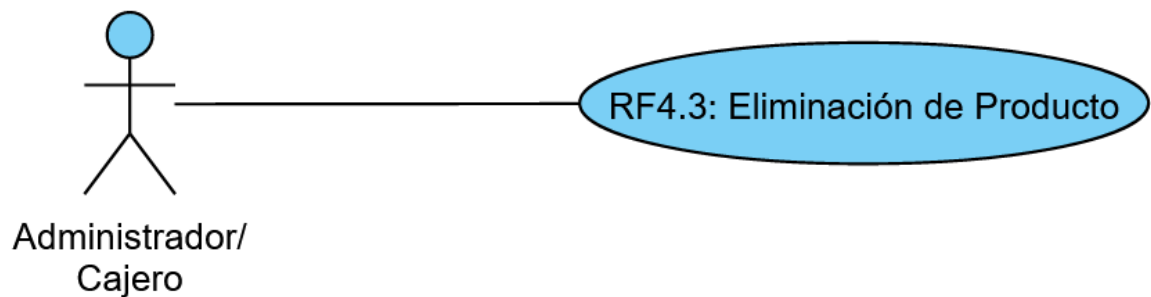
Excepciones

- E1 Si los datos del Producto están incompletos.
- E2 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF4.3 Eliminación de Producto

Figura 23

Diagrama de Caso de Uso A Detalle - Modificación de Producto



Autor: Cadena, S. (2023)

Actores: Administrador, Encargado Producto

Flujo Principal

10. El actor presiona la opción de Gestionar Producto.

11. El sistema despliega formulario de Gestión de Producto.
12. El actor selecciona la opción Eliminación de Producto.
13. El sistema despliega pantalla para Eliminación Producto.
14. El actor ingresa los datos necesarios.
15. El actor presiona Eliminar Producto.
16. El sistema desplegará un mensaje: "¿Está seguro de Eliminar el Producto?".
17. El usuario Selecciona Si
18. El sistema Eliminará la información (E1)

Flujo Alternativo

1. Puede seleccionar RF1.1 Ingreso de Producto, RF1.2 Modificación de Producto o RF1.4 Consulta de Producto
8. El usuario selecciona No, el sistema no Eliminará la información

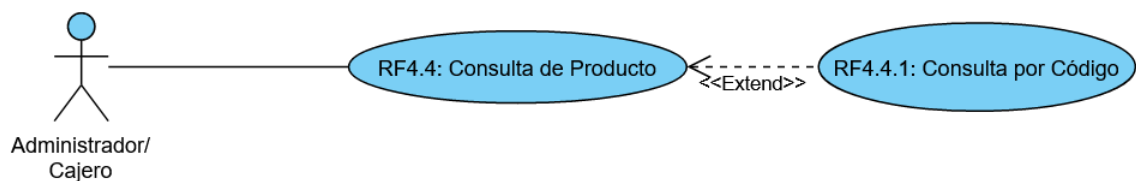
Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF4.4.1 Consulta de Producto por Código

Figura 24

Diagrama de Caso de Uso A Detalle - Consulta de Producto por Código



Autor: Cadena, S. (2023)

Actores: Administrador, Encargado Inventario

Flujo Principal

1. El actor presiona la opción de Gestionar Producto.
2. El sistema despliega formulario de Gestión de Producto.
3. El actor selecciona la opción Consulta de Producto.
4. El sistema despliega pantalla para Consulta de Producto.
5. El actor selecciona la opción Consulta de Producto por Código.
6. El sistema presenta la pantalla para Consulta de Producto por Código.
7. El actor ingresa el Nombre que desea consultar.
8. El sistema verifica el Código.
9. El sistema desplegará información del Producto Consultado (E1)

Flujo Alternativo

1. Puede seleccionar RF4.1 Ingreso de Producto, RF4.2 Modificación de Producto o RF4.3 Eliminación de Producto.
5. El usuario selecciona Consulta de Producto por Categoría.

Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF4.4.2 Consulta de Producto por Nombre

Figura 25

Diagrama de Caso de Uso A Detalle - Consulta de Producto por Nombre



Autor: Cadena, S. (2023)

Actores: Administrador, Encargado Inventario

Flujo Principal

10. El actor presiona la opción de Gestionar Producto.
11. El sistema despliega formulario de Gestión de Producto.
12. El actor selecciona la opción Consulta de Producto.
13. El sistema despliega pantalla para Consulta de Producto.
14. El actor selecciona la opción Consulta de Producto por Código.
15. El sistema presenta la pantalla para Consulta de Producto por Código.
16. El actor ingresa el Nombre que desea consultar.
17. El sistema verifica el Código.
18. El sistema desplegará información del Producto Consultado (E1)

Flujo Alternativo

2. Puede seleccionar RF4.1 Ingreso de Producto, RF4.2 Modificación de Producto o RF4.3 Eliminación de Producto.
6. El usuario selecciona Consulta de Producto por Categoría.

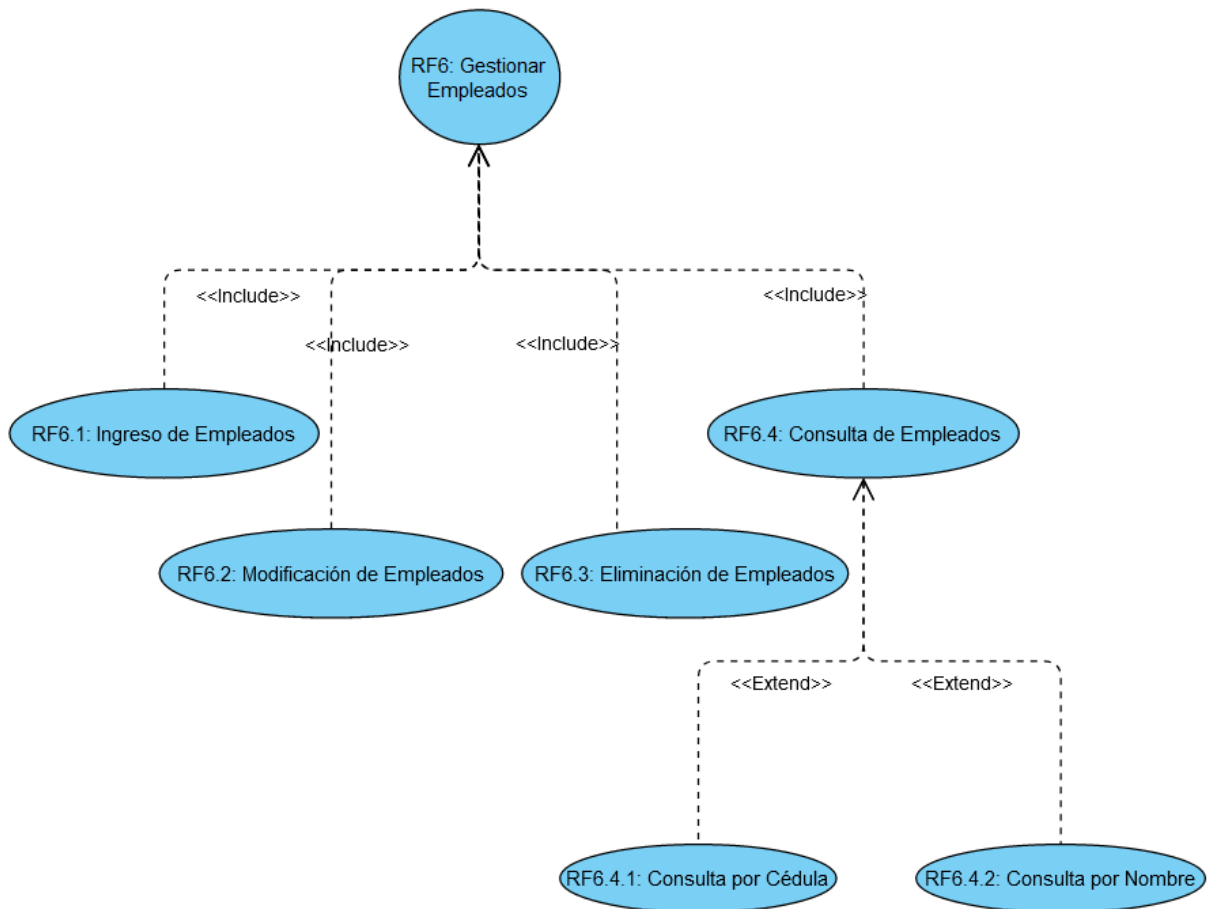
Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

4.3.2.5. RF6 Gestionar Empleados

Figura 26

Diagrama de Caso de Uso Siguiete Nivel - Gestionar Empleado

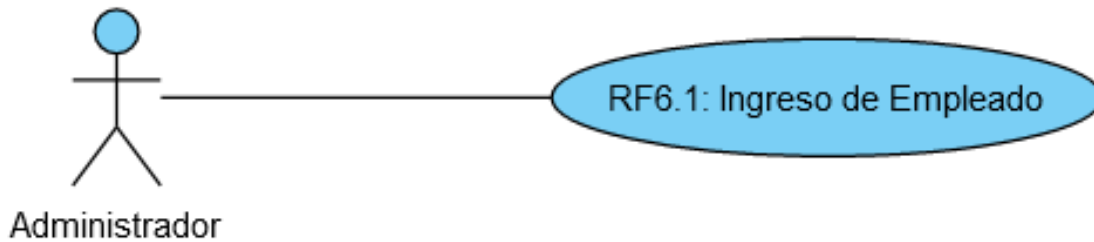


Autor: Cadena, S. (2023)

ID: RF6.1 Ingreso de Empleado

Figura 27

Diagrama de Caso de Uso A Detalle - Ingreso de Empleado



Actores: Administrador

Flujo Principal

1. El actor presiona la opción de Gestionar Empleado.
2. El sistema despliega formulario de Gestión de Empleado.
3. El actor selecciona la opción Ingreso de Empleado.
4. El sistema despliega pantalla para Ingresar Empleado.
5. El actor ingresa los datos necesarios de un nuevo Empleado.
6. El actor presiona Guardar.
7. El sistema verifica su existencia (E1).
8. El sistema almacena la información (E2).

Flujo Alternativo

1. Puede seleccionar RF6.2 Modificación de Empleado, RF6.3 Eliminación de Empleado o RF6.4 Consulta de Empleado.

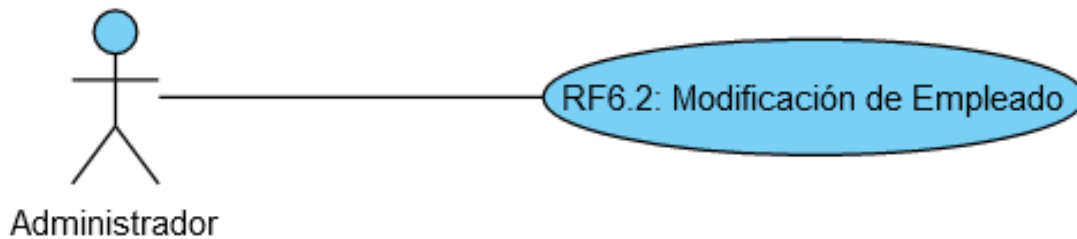
Excepciones

- E1 Si el Empleado ya existe el sistema presentará un mensaje el cual indique que el Empleado ya existe.
- E2 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF6.2 Modificación de Empleado

Figura 28

Diagrama de Caso de Uso A Detalle - Modificación de Empleados



Autor: Cadena, S. (2023)

Actores: Administrador

Flujo Principal

1. El actor presiona la opción de Gestionar Empleado.
2. El sistema despliega formulario de Gestión de Empleado.
3. El actor selecciona la opción Modificación de Empleado.
4. El sistema despliega pantalla para modificar Empleado.
5. El actor ingresa los datos necesarios para modificar el Empleado.
6. El actor presiona Guardar.
7. El sistema Verifica la Información. (E1)
8. El sistema almacena la información. (E2)

Flujo Alternativo

1. Puede seleccionar RF6.1 Modificación de Empleado, RF6.3 Eliminación de Empleado o RF6.4 Consulta de Empleado

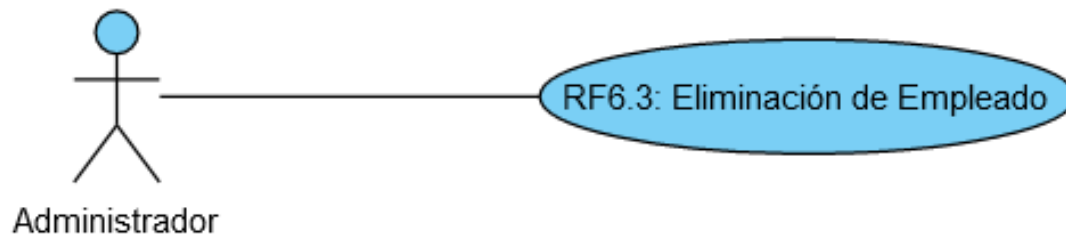
Excepciones

- E1 Si los datos del Empleado están incompletos.
- E2 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF6.3 Eliminación de Empleado

Figura 29

Diagrama de Caso de Uso A Detalle - Eliminación de Empleado



Autor: Cadena, S. (2023)

Actores: Administrador

Flujo Principal

1. El actor presiona la opción de Gestionar Empleado. El sistema despliega formulario de Gestión de Empleado.
2. El actor selecciona la opción Eliminación de Empleado.
3. El sistema despliega pantalla para Eliminación Empleado.
4. El actor ingresa los datos necesarios.
5. El actor presiona Eliminar Empleado.

6. El sistema desplegará un mensaje: "¿Está seguro de Eliminar el Empleado?".
7. El usuario Selecciona Si
8. El sistema Eliminará la información (E1)

Flujo Alternativo

1. Puede seleccionar RF1.1 Ingreso de Empleado, RF1.2 Modificación de Empleado o RF1.4 Consulta de Empleado.
8. El usuario selecciona No, el sistema no Eliminará la información

Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF6.4.1 Consulta de Empleado por Cédula

Figura 30

Diagrama de Caso de Uso A Detalle - Consulta de Empleados por Cédula



Autor: Cadena, S. (2023)

Actores: Administrador

Flujo Principal

1. El actor presiona la opción de Gestionar Empleado.
2. El sistema despliega formulario de Gestión de Empleado.

3. El actor selecciona la opción Consulta de Empleado.
4. El sistema despliega pantalla para Consulta de Empleado.
5. El actor selecciona la opción Consulta de Empleado por Cédula.
6. El sistema presenta la pantalla para Consulta de Empleado por Cédula.
7. El actor ingresa la Categoría que desea consultar.
8. El sistema verifica el Cédula.
9. El sistema desplegará información del Empleado Consultado (E1)

Flujo Alternativo

1. Puede seleccionar RF6.1 Ingreso de Empleado, RF6.2 Modificación de Empleado o RF6.3 Eliminación de Empleado
5. El usuario selecciona Consulta de Empleado por Nombre

Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF6.4.2 Consulta de Empleado por Nombre

Figura 31

Diagrama de Caso de Uso A Detalle - Consulta de Empleado por Nombre



Autor: Cadena, S. (2023)

Actores: Administrador

Flujo Principal

10. El actor presiona la opción de Gestionar Empleado.
11. El sistema despliega formulario de Gestión de Empleado.
12. El actor selecciona la opción Consulta de Empleado.
13. El sistema despliega pantalla para Consulta de Empleado.
14. El actor selecciona la opción Consulta de Empleado por Nombre.
15. El sistema presenta la pantalla para Consulta de Empleado por Nombre.
16. El actor ingresa la Nombre que desea consultar.
17. El sistema verifica el Nombre.
18. El sistema desplegará información del Empleado Consultado (E1)

Flujo Alternativo

2. Puede seleccionar RF6.1 Ingreso de Empleado, RF6.2 Modificación de Empleado o RF6.3 Eliminación de Empleado
6. El usuario selecciona Consulta de Empleado por Cédula

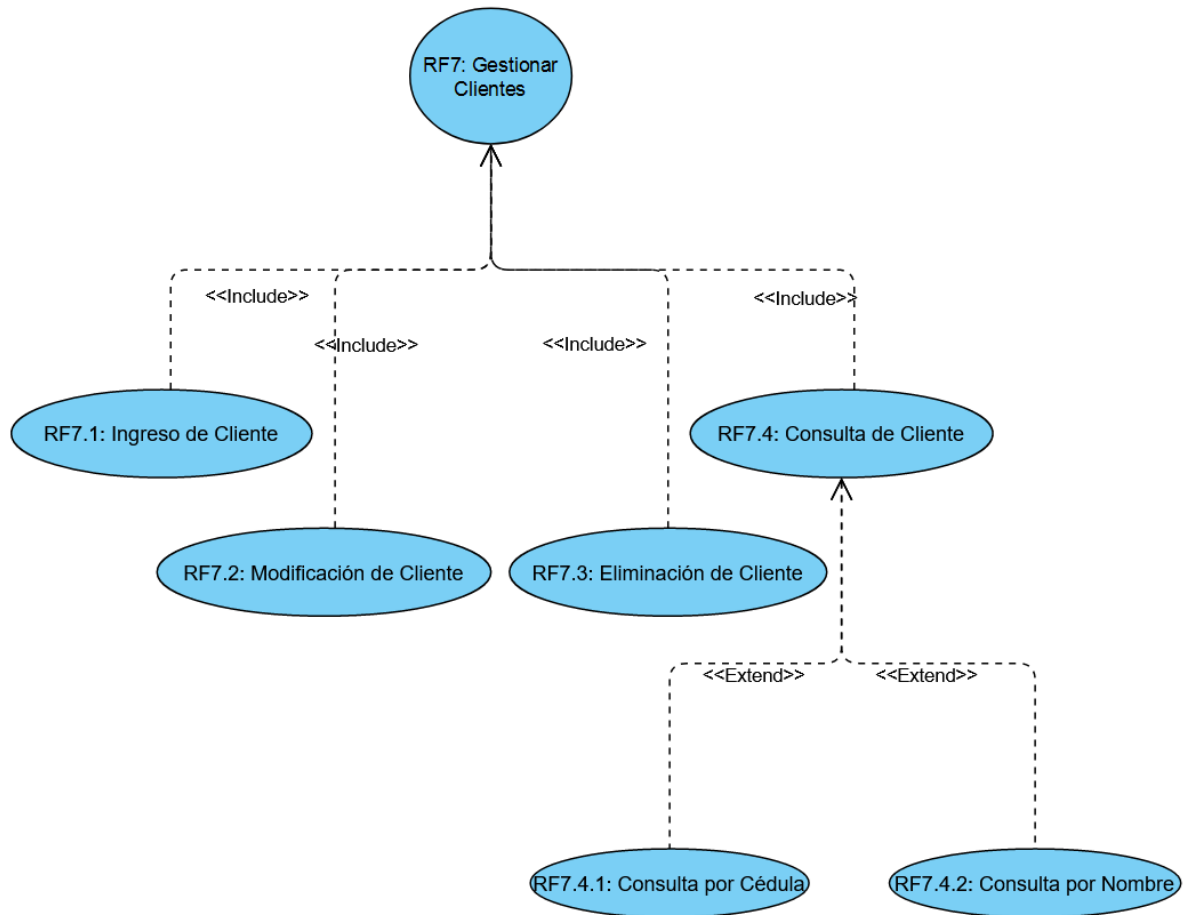
Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador.

4.3.2.6. RF7 Gestionar Cliente

Figura 32

Diagrama de Caso de Uso Siguiente Nivel - Gestionar Cliente

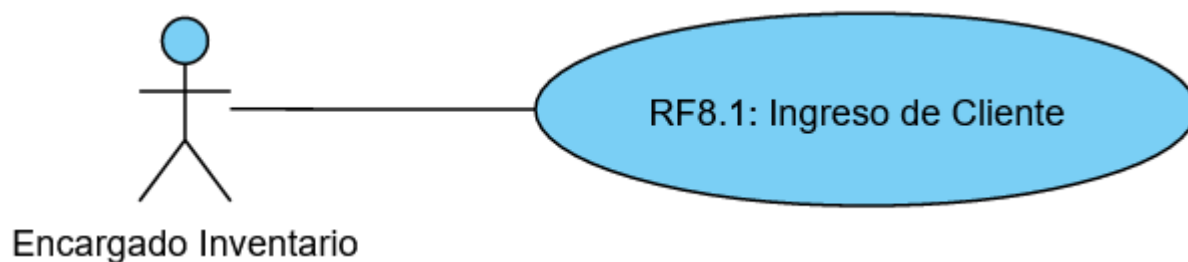


Autor: Cadena, S. (2023)

ID: RF7.1 Ingreso de Cliente

Figura 33

Diagrama de Caso de Uso A Detalle - Ingreso de Cliente



Autor: Cadena, S. (2023)

Actores: Encargado de Cocina

Flujo Principal

1. El actor presiona la opción de Gestionar Cliente.
2. El sistema despliega formulario de Gestión de Cliente.
3. El actor selecciona la opción Ingreso de Cliente.
4. El sistema despliega pantalla para Ingresar Cliente.
5. El actor ingresa los datos necesarios de un nuevo Cliente.
6. El actor presiona Guardar.
7. El sistema verifica su existencia (E1).
8. El sistema almacena la información (E2).

Flujo Alternativo

1. Puede seleccionar RF1.2 Modificación de Cliente, RF1.3 Eliminación de Cliente o RF1.4 Consulta de Cliente.

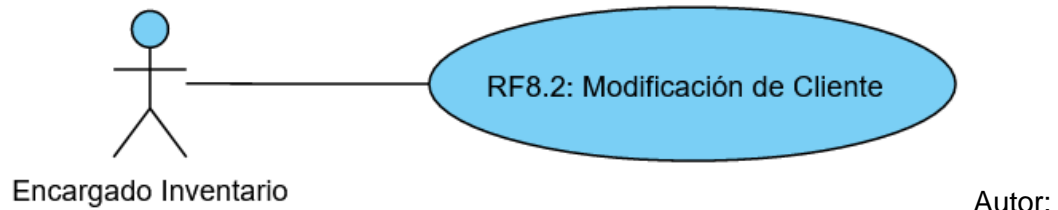
Excepciones

- E1 Si el Cliente ya existe el sistema presentará un mensaje el cual indique que el Cliente ya existe.
- E2 Existen problemas de conexión con la base de datos, se requiere del administrador.

ID: RF7.2 Modificación de Cliente

Figura 34

Diagrama de Caso de Uso A Detalle - Modificación de Cliente



Cadena, S. (2023)

Actores: Encargado de Cocina

Flujo Principal

1. El actor presiona la opción de Gestionar Cliente.
2. El sistema despliega formulario de Gestión de Cliente.
3. El actor selecciona la opción Modificación de Cliente.
4. El sistema despliega pantalla para modificar Cliente.
5. El actor ingresa los datos necesarios para modificar el Cliente.
6. El actor presiona Guardar.
7. El sistema Verifica la Información. (E1)
8. El sistema almacena la información. (E2)

Flujo Alterno

2. Puede seleccionar RF7.1 Modificación de Cliente, RF7.3 Eliminación de Cliente o RF7.4 Consulta de Cliente

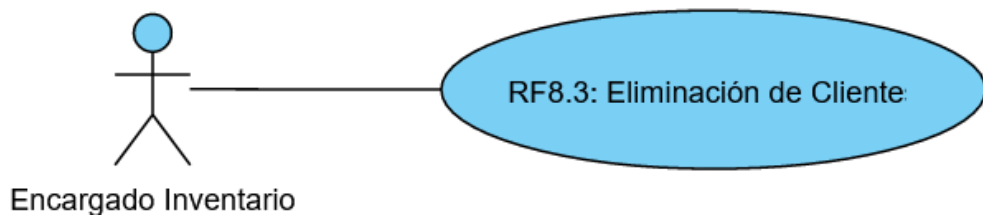
Excepciones

- E1 Si los datos del Cliente están incompletos.
- E2 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF7.3 Eliminación de Cliente

Figura 35

Diagrama de Caso de Uso A Detalle - Eliminación de Cliente



Autor:

Cadena, S. (2023)

Actores: Encargado de Cocina

Flujo Principal

1. El actor presiona la opción de Gestionar Cliente.
2. El sistema despliega formulario de Gestión de Cliente.
3. El actor selecciona la opción Eliminación de Cliente.
4. El sistema despliega pantalla para Eliminación Cliente.
5. El actor ingresa los datos necesarios.

6. El actor presiona Eliminar Cliente.
7. El sistema desplegará un mensaje: "¿Está seguro de Eliminar el Cliente?".
8. El usuario Selecciona Si
9. El sistema Eliminará la información (E1)

Flujo Alternativo

1. Puede seleccionar RF7.1 Ingreso de Cliente, RF7.2 Modificación de Cliente o RF7.4 Consulta de Cliente
8. El usuario selecciona No, el sistema no Eliminará la información

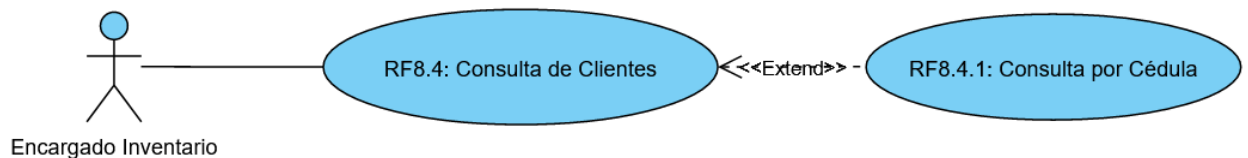
Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF7.4.1 Consulta de Cliente por Cédula

Figura 36

Diagrama de Caso de Uso A Detalle - Consulta de Clientes por Cédula



Autor: Cadena, S. (2023)

Actores: Encargado de Cocina

Flujo Principal

1. El actor presiona la opción de Gestionar Cliente.
2. El sistema despliega formulario de Gestión de Cliente.

3. El actor selecciona la opción Consulta de Cliente.
4. El sistema despliega pantalla para Consulta de Cliente.
5. El actor selecciona la opción Consulta de Cliente por Cédula.
6. El sistema presenta la pantalla para Consulta de Cliente por Cédula.
7. El actor ingresa el nombre que desea consultar.
8. El sistema verifica el Cédula.
9. El sistema desplegará información del Cliente Consultado (E1)

Flujo Alternativo

1. Puede seleccionar RF7.1 Ingreso de Cliente, RF7.2 Modificación de Cliente o RF7.3 Eliminación de Cliente
5. El usuario selecciona Consulta de Cliente por Nombre

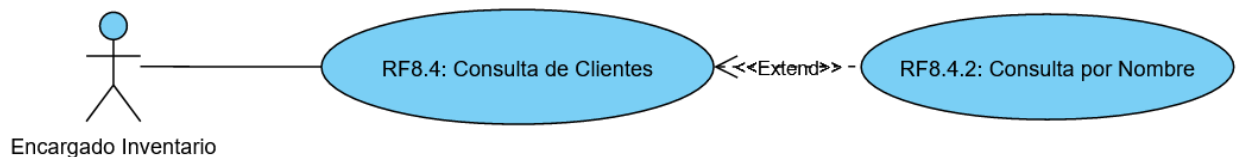
Excepciones

E1 Existen problemas de conexión con la base de datos, se requiere del administrador

ID: RF7.4.2 Consulta de Cliente por Nombre

Figura 37

Diagrama de Caso de Uso A Detalle - Consulta de Clientes por Nombre



Autor: Cadena, S. (2023)

Actores: Encargado de Cocina

Flujo Principal

1. El actor presiona la opción de Gestionar Cliente.
2. El sistema despliega formulario de Gestión de Cliente.
3. El actor selecciona la opción Consulta de Cliente.
4. El sistema despliega pantalla para Consulta de Cliente.
5. El actor selecciona la opción Consulta de Cliente por Nombre.
6. El sistema presenta la pantalla para Consulta de Cliente por Nombre.
7. El actor ingresa el nombre que desea consultar.
8. El sistema verifica el nombre.
9. El sistema desplegará información del Cliente Consultado (E1)

Flujo Alternativo

1. Puede seleccionar RF7.1 Ingreso de Cliente, RF7.2 Modificación de Cliente o RF7.3 Eliminación de Cliente
5. El usuario selecciona Consulta de Cliente por Nombre

Excepciones

- E1 Existen problemas de conexión con la base de datos, se requiere del administrador

4.3.3. Diagramas de Secuencia

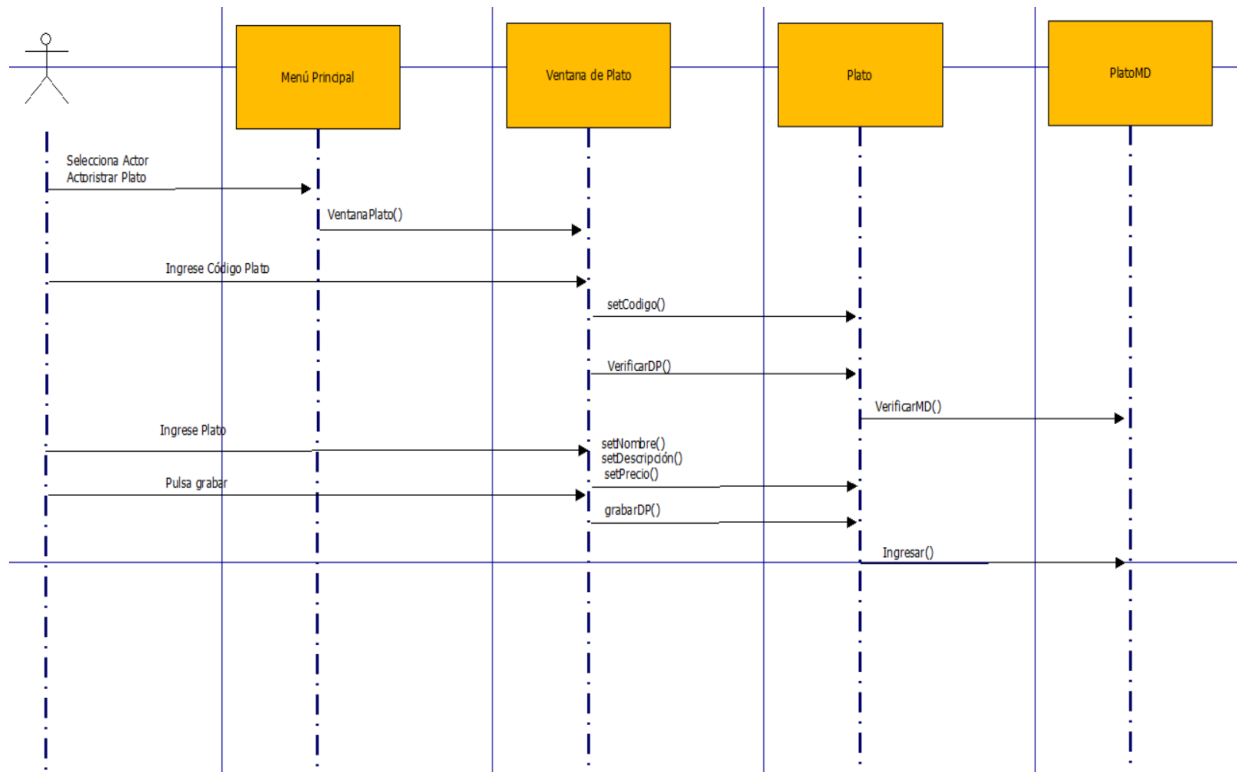
4.3.3.1. RF1 Gestionar Plato

Crear Plato:

El diagrama de secuencia del requerimiento funcional Gestionar Platos se centra en describir claramente los pasos que debe seguir el usuario para ingresar un plato.

Figura 38

Diagrama de secuencia crear plato



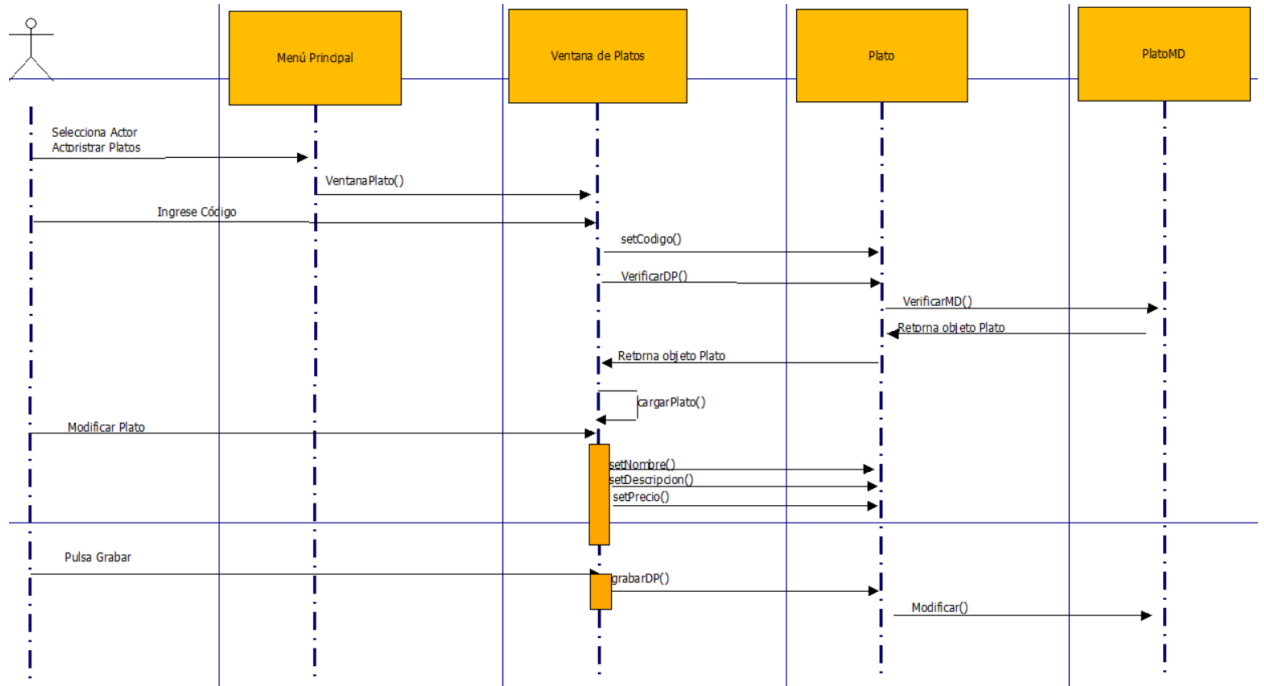
Autor: Cadena, S. (2023)

Modificar Plato:

El diagrama de secuencia del requerimiento funcional Gestionar Plato se centra en describir claramente los pasos que debe seguir el usuario para modificar un plato.

Figura 39

Diagrama de secuencia modificar plato



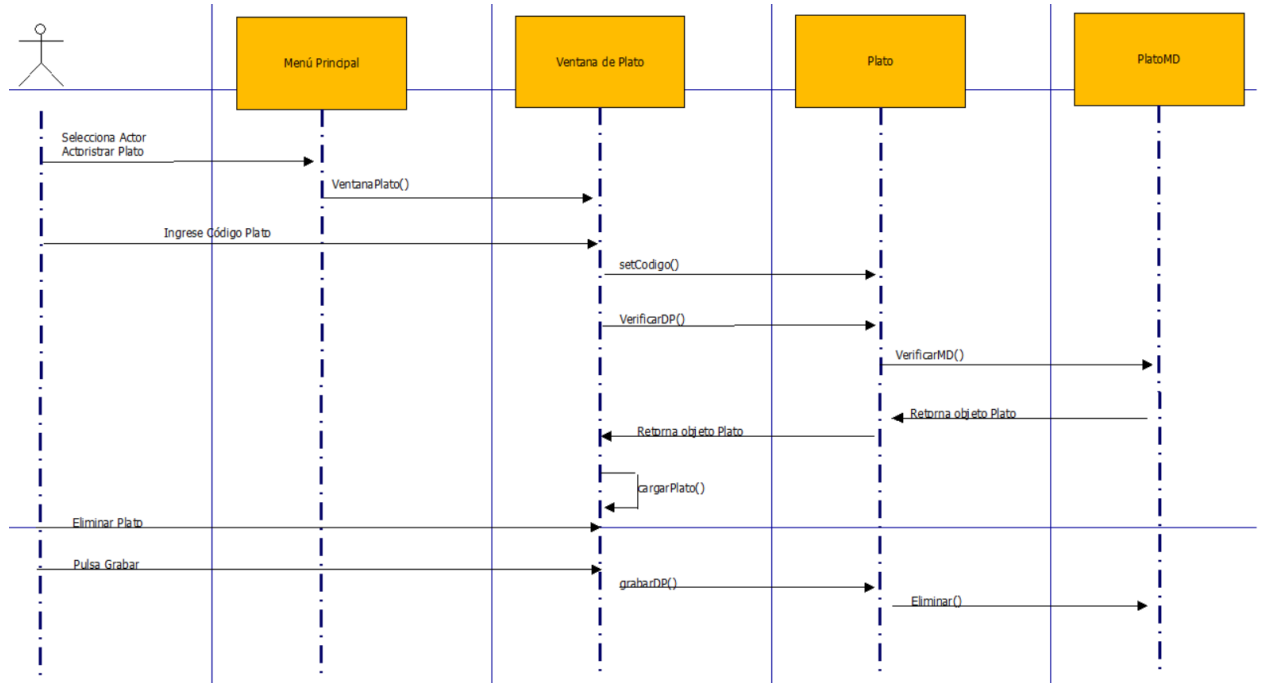
Autor: Cadena, S. (2023)

Eliminar Plato:

El diagrama de secuencia del requerimiento funcional Gestionar Platos se centra en describir claramente los pasos que debe seguir el usuario para eliminar un plato.

Figura 40

Diagrama de secuencia eliminar plato



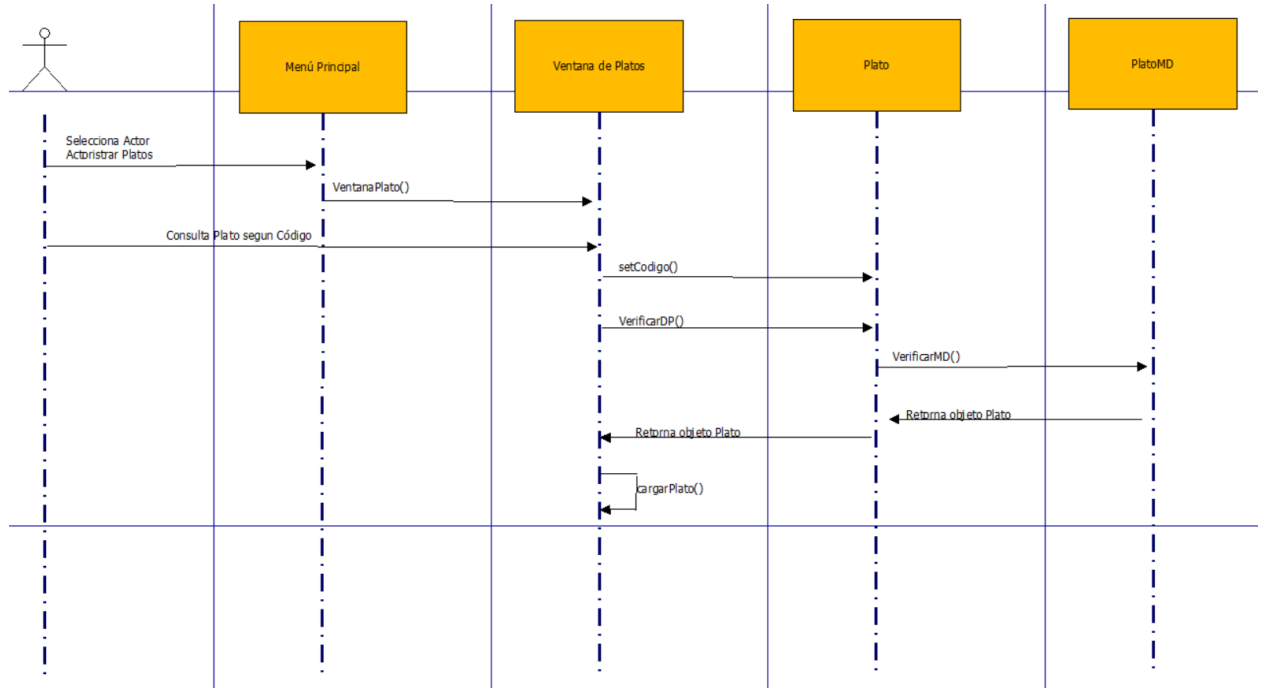
Autor: Cadena, S. (2023)

Buscar Plato por Código:

El diagrama de secuencia del requerimiento funcional Gestionar Platos se centra en describir claramente los pasos que debe seguir el usuario para buscar un plato por su código.

Figura 41

Diagrama de secuencia buscar plato por código



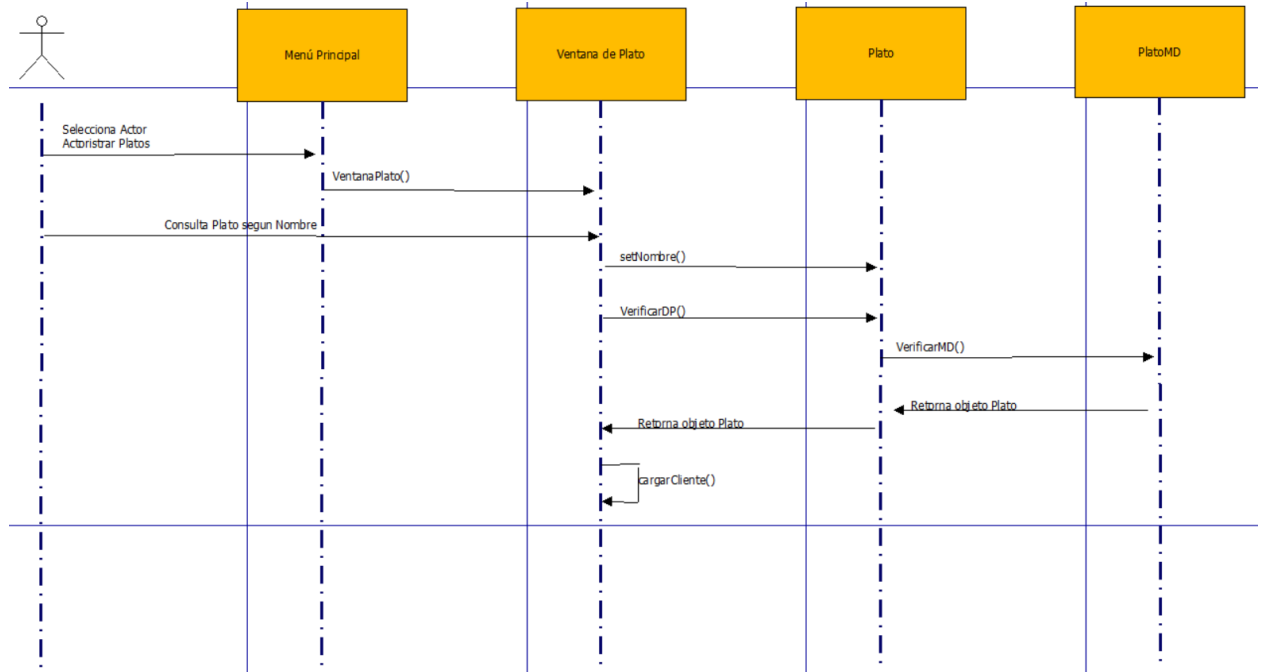
Autor: Cadena, S. (2023)

Buscar Plato por Nombre:

El diagrama de secuencia del requerimiento funcional Gestionar Platos se centra en describir claramente los pasos que debe seguir el usuario para buscar un plato por su nombre.

Figura 42

Diagrama de secuencia buscar plato por nombre



Autor: Cadena, S. (2023)

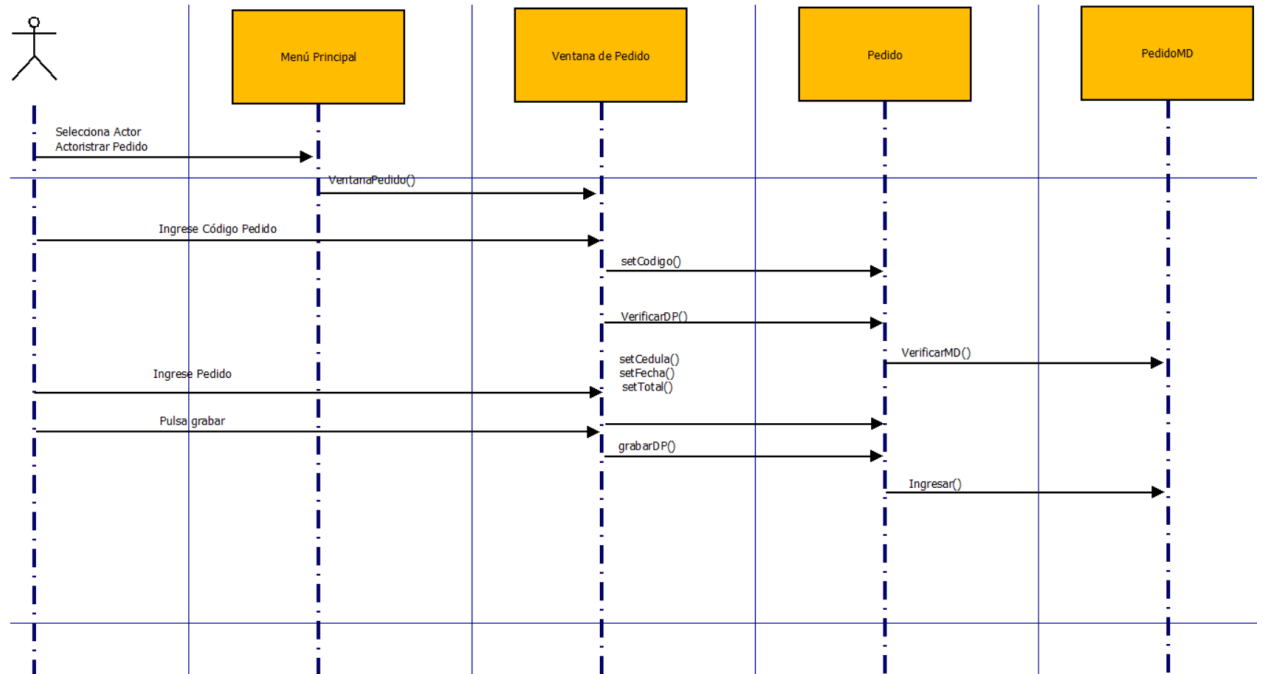
4.3.3.2. RF2 Gestionar Pedido

Crear Pedido:

El diagrama de secuencia del requerimiento funcional Gestionar Pedidos se centra en describir claramente los pasos que debe seguir el usuario para crear un pedido.

Figura 43

Diagrama de secuencia crear pedido



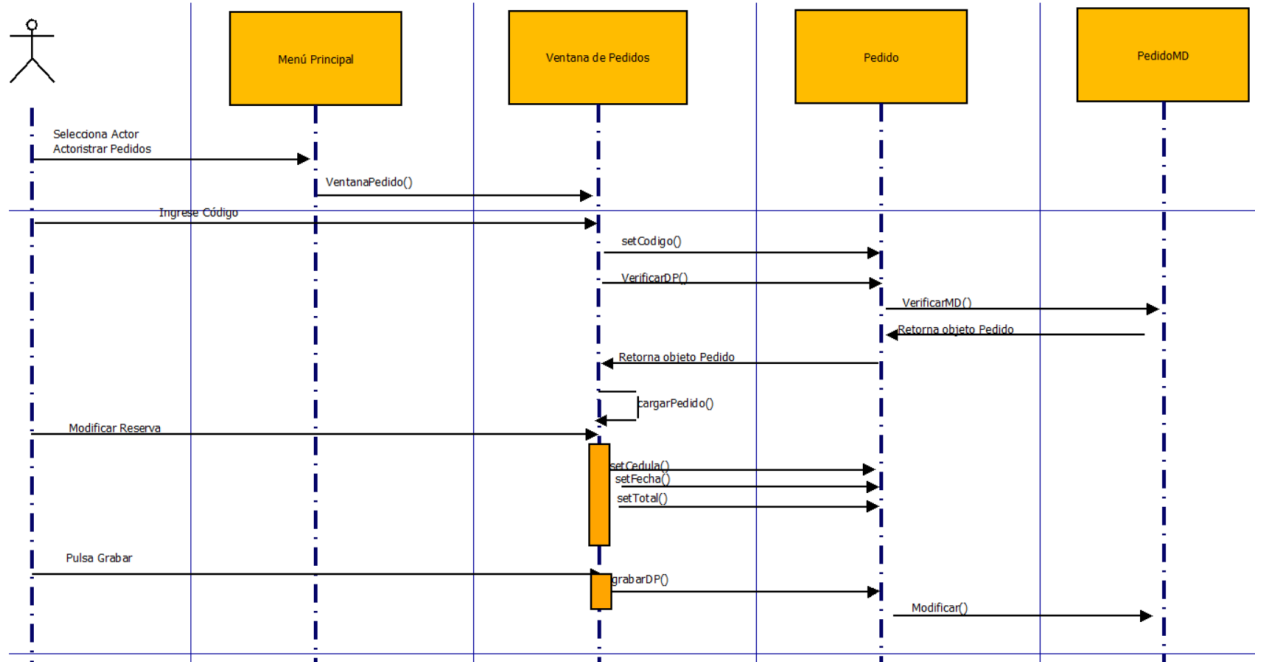
Autor: Cadena, S. (2023)

Modificar Pedido:

El diagrama de secuencia del requerimiento funcional Gestionar Platos se centra en describir claramente los pasos que debe seguir el usuario para modificar un pedido.

Figura 44

Diagrama de secuencia modificar pedido



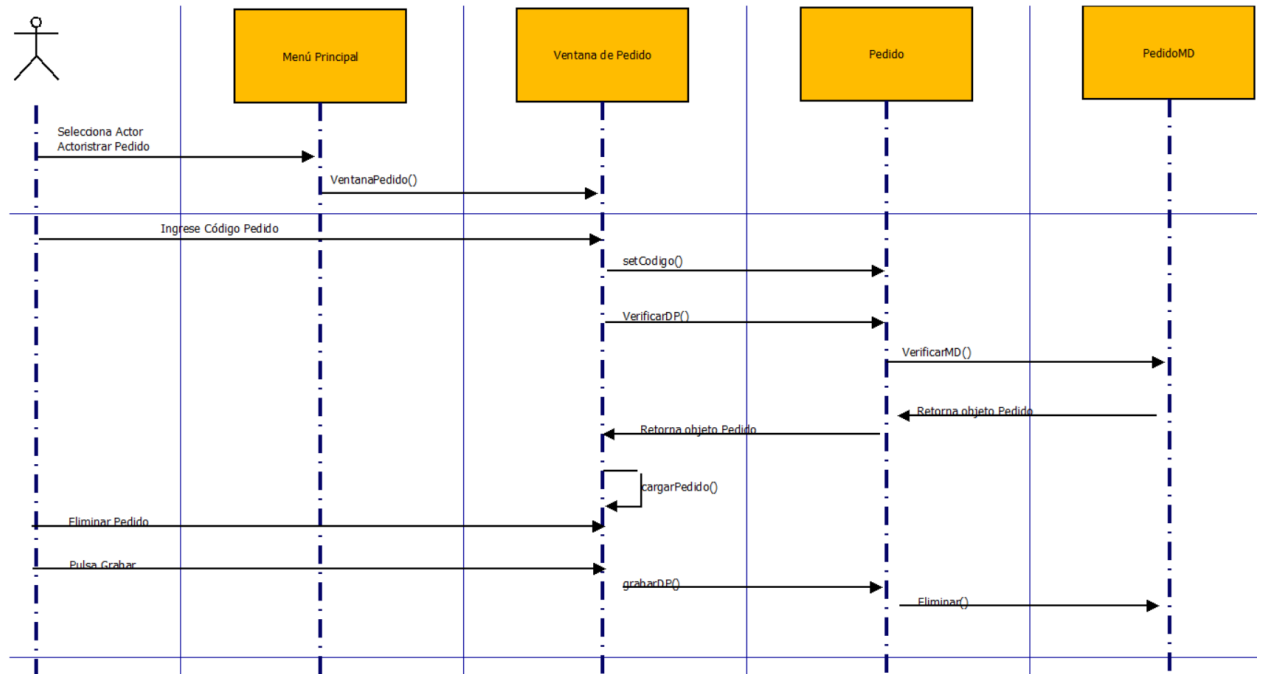
Autor: Cadena, S. (2023)

Eliminar Pedido:

El diagrama de secuencia del requerimiento funcional Gestionar Platos se centra en describir claramente los pasos que debe seguir el usuario para eliminar un pedido.

Figura 45

Diagrama de secuencia eliminar pedido



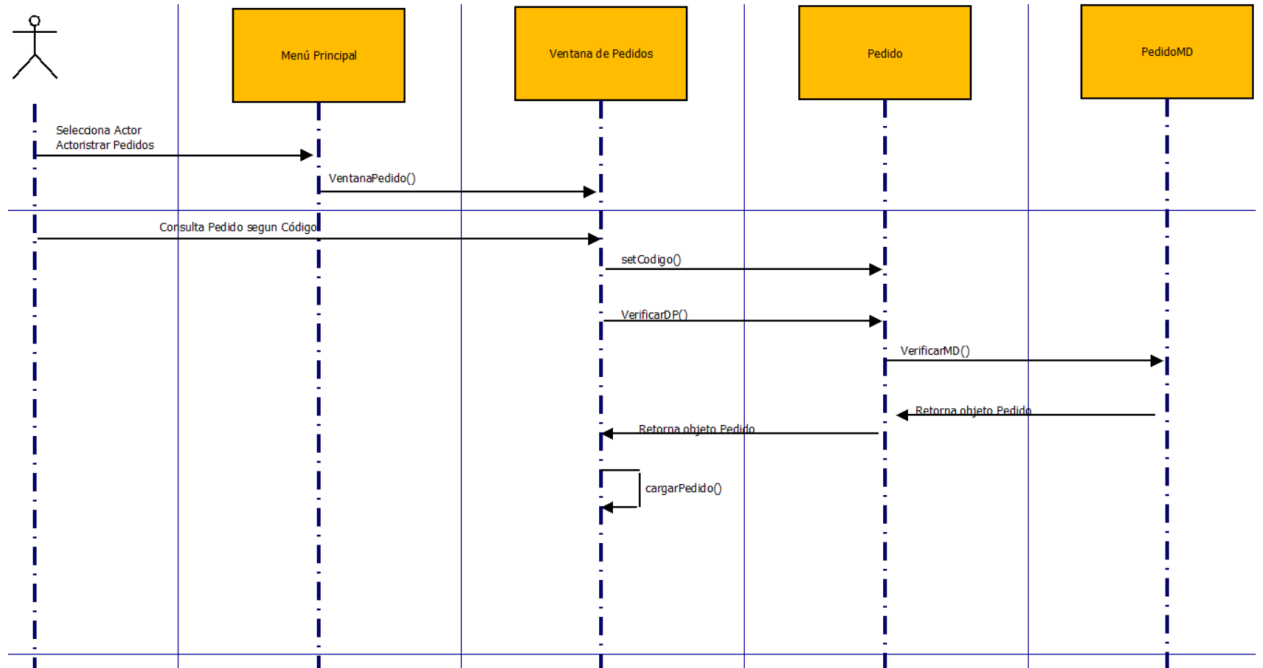
Autor: Cadena, S. (2023)

Buscar Pedido por Código:

El diagrama de secuencia del requerimiento funcional Gestionar Platos se centra en describir claramente los pasos que debe seguir el usuario para buscar un pedido por el código del pedido.

Figura 46

Diagrama de secuencia buscar pedido por código



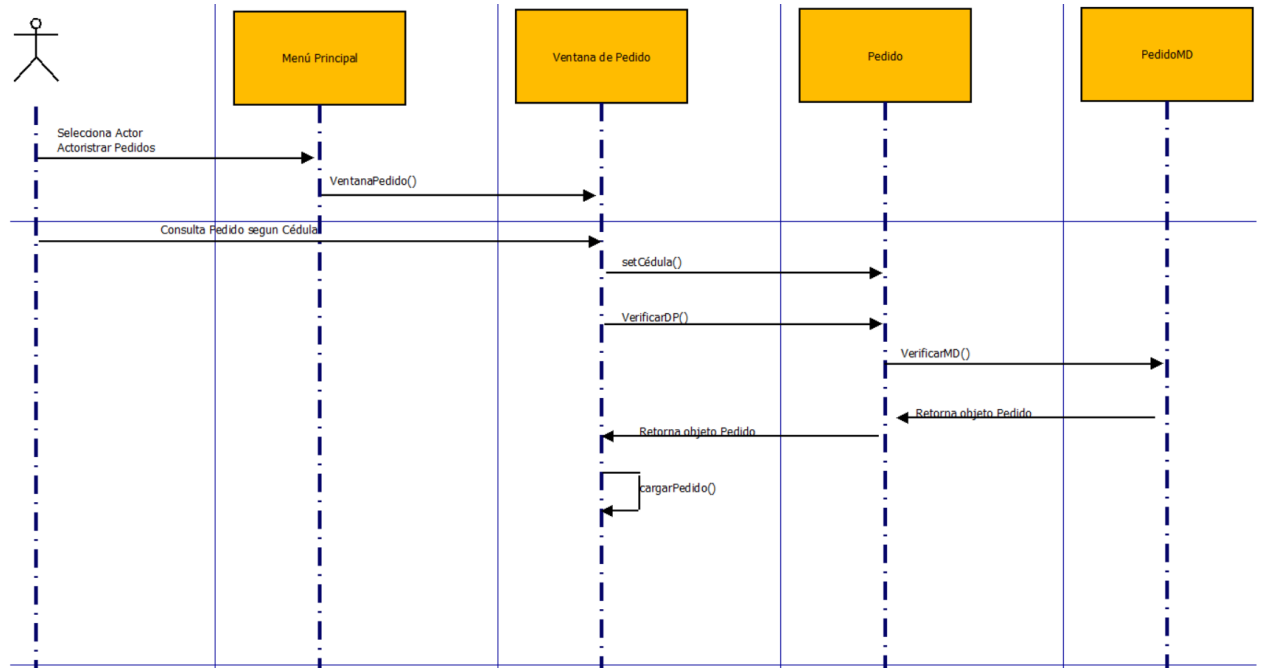
Autor: Cadena, S. (2023)

Buscar Pedido por Cédula:

El diagrama de secuencia del requerimiento funcional Gestionar Platos se centra en describir claramente los pasos que debe seguir el usuario para buscar un pedido por la Cédula del cliente.

Figura 47

Diagrama de secuencia buscar pedido por cédula



Autor: Cadena, S. (2023)

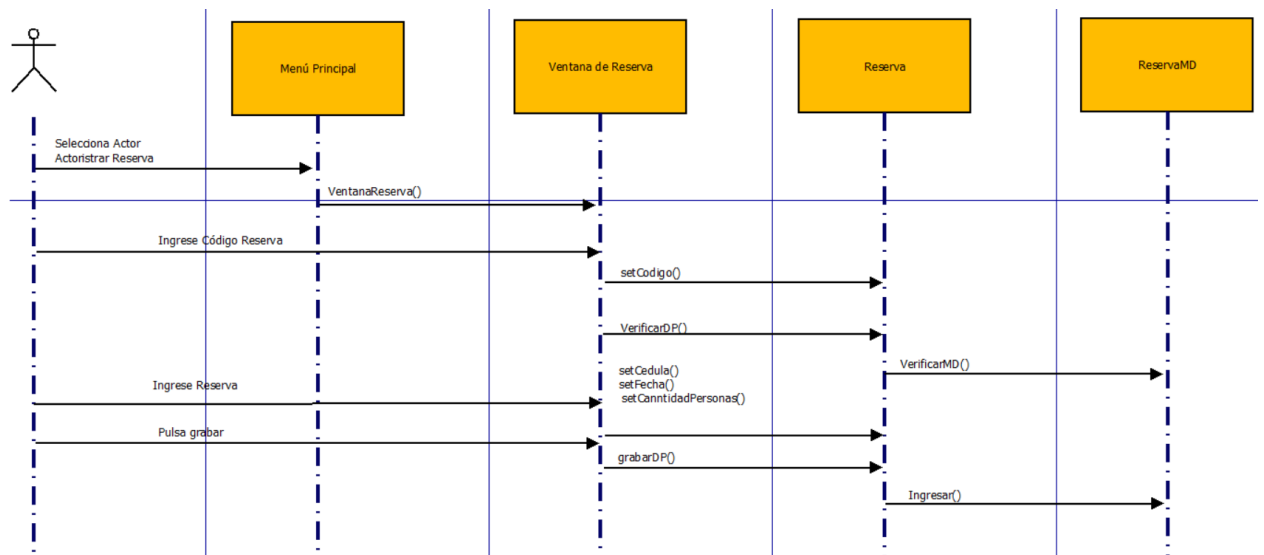
4.3.3.3. RF3 Gestionar Reservas

Crear Reserva:

El diagrama de secuencia del requerimiento funcional Gestionar Reservas se centra en describir claramente los pasos que debe seguir el usuario para crear una reserva.

Figura 48

Diagrama de secuencia crear reserva



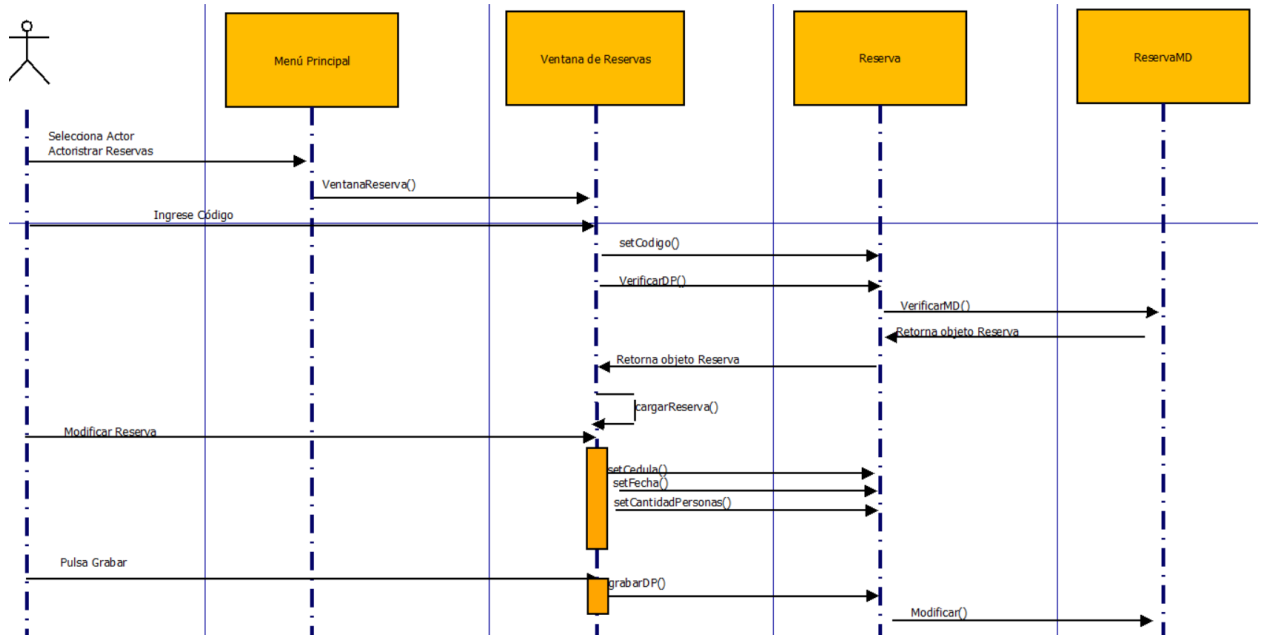
Autor: Cadena, S. (2023)

Modificar Reserva:

El diagrama de secuencia del requerimiento funcional Gestionar Reservas se centra en describir claramente los pasos que debe seguir el usuario para modificar una reserva.

Figura 49

Diagrama de secuencia modificar reserva



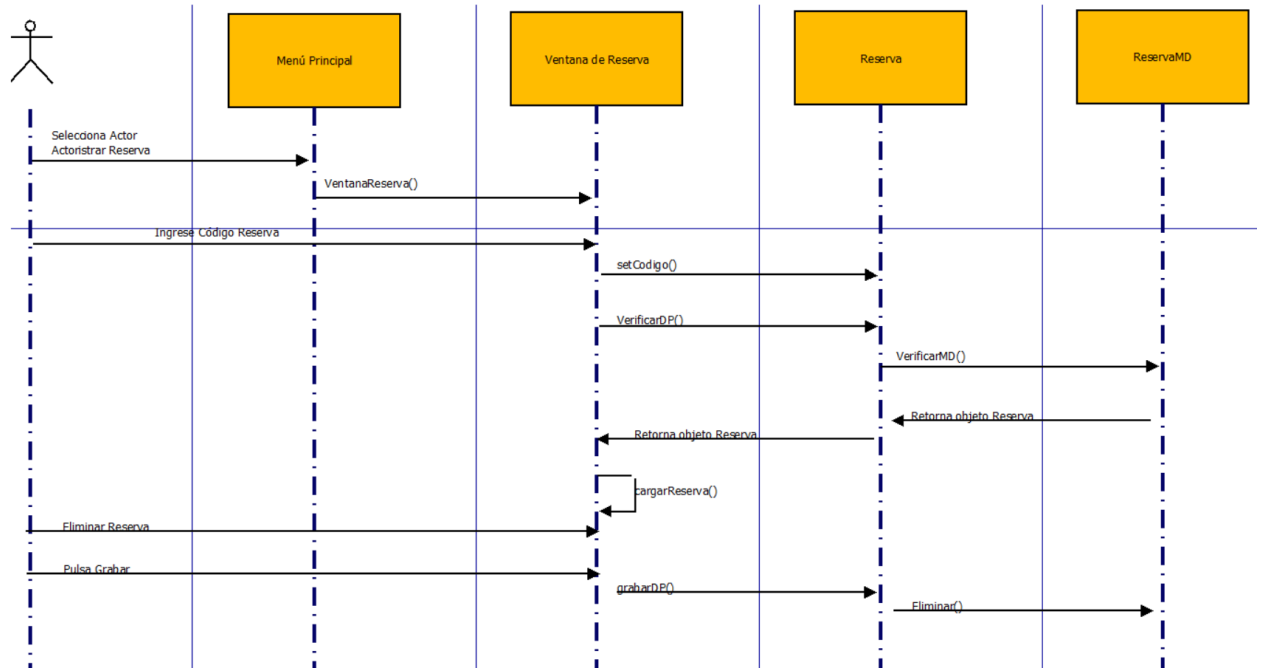
Autor: Cadena, S. (2023)

Eliminar Reserva:

El diagrama de secuencia del requerimiento funcional Gestionar Reservas se centra en describir claramente los pasos que debe seguir el usuario para eliminar una reserva.

Figura 50

Diagrama de secuencia eliminar reserva



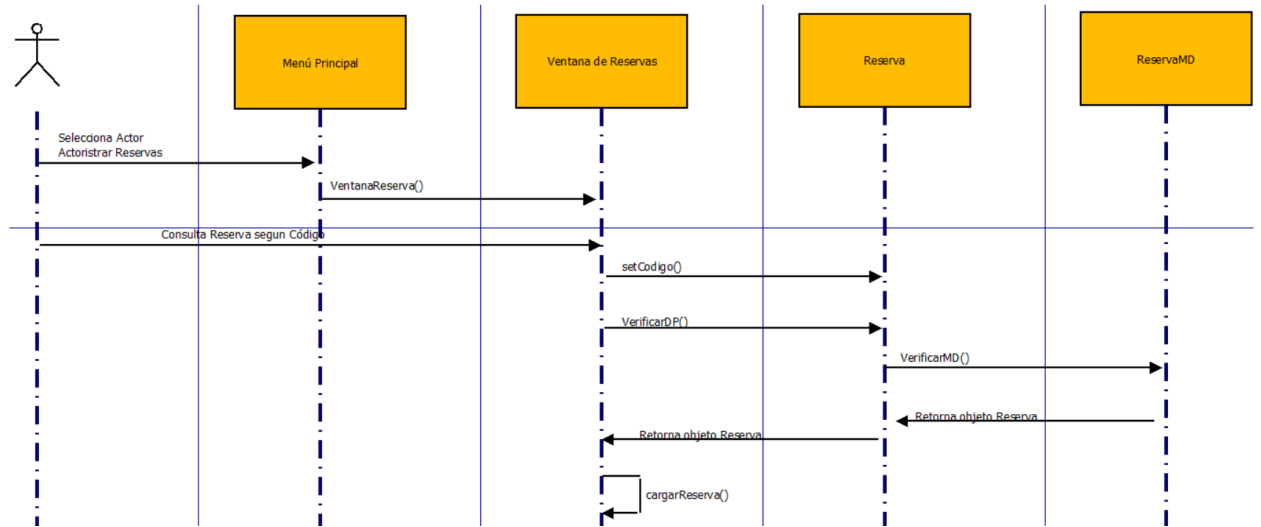
Autor: Cadena, S. (2023)

Buscar Reservas por Código:

El diagrama de secuencia del requerimiento funcional Gestionar Reservas se centra en describir claramente los pasos que debe seguir el usuario para buscar una reserva por su código de reserva.

Figura 51

Diagrama de secuencia buscar reserva por código



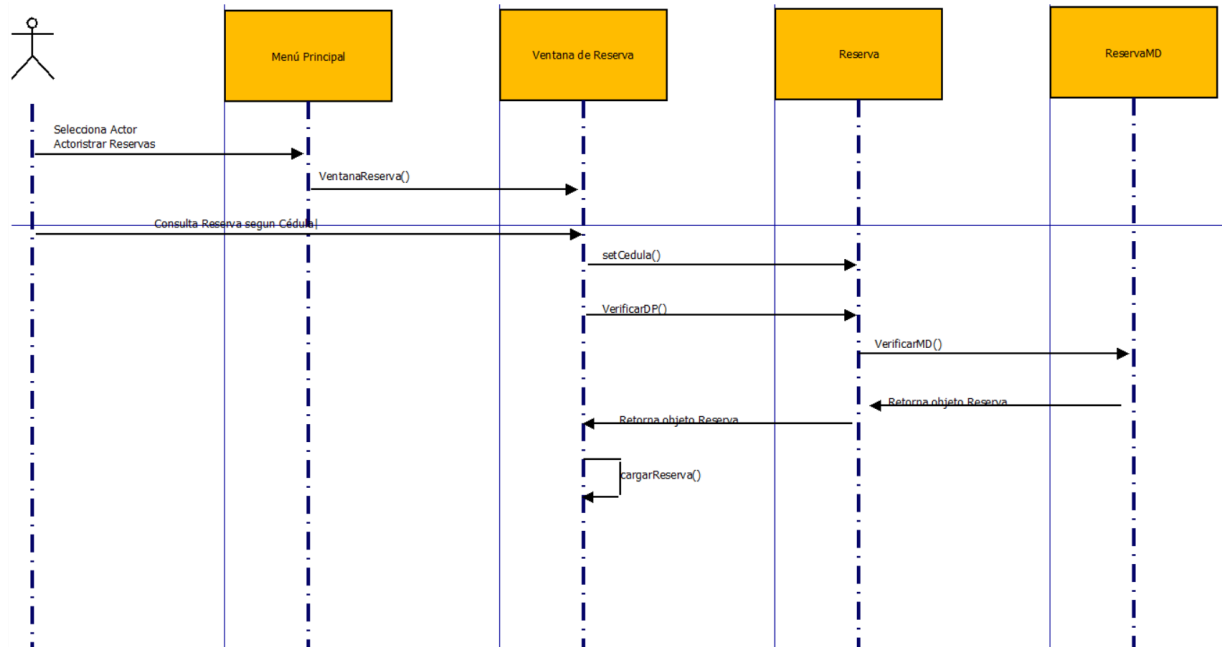
Autor: Cadena, S. (2023)

Buscar Reservas por Cédula:

El diagrama de secuencia del requerimiento funcional Gestionar Reservas se centra en describir claramente los pasos que debe seguir el usuario para buscar una reserva por la cédula del cliente.

Figura 52

Diagrama de secuencia buscar reserva por cédula



Autor: Cadena, S. (2023)

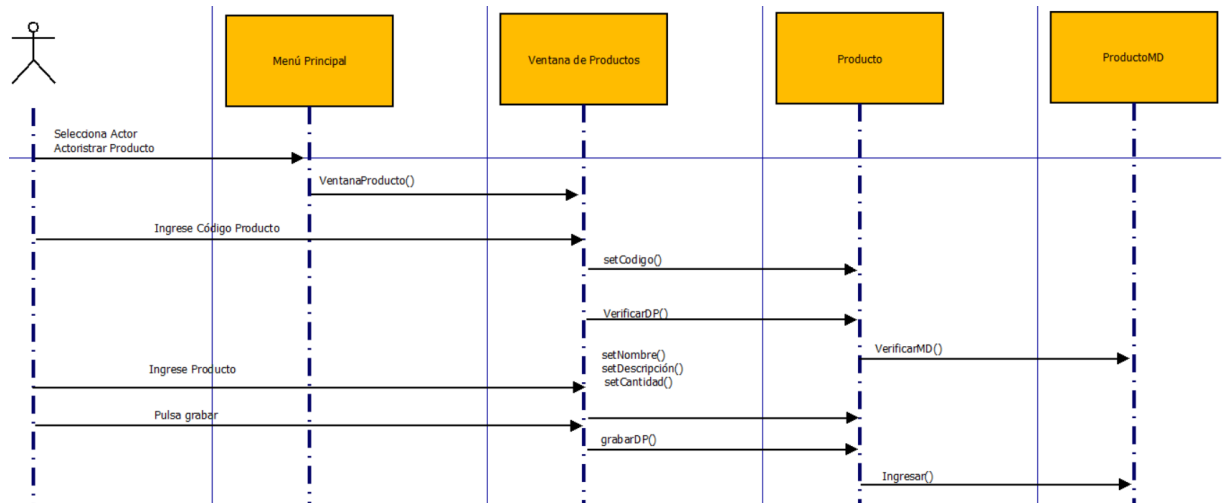
4.3.3.4. RF4 Gestionar Producto

Crear Producto:

El diagrama de secuencia del requerimiento funcional Gestionar Productos se centra en describir claramente los pasos que debe seguir el usuario para ingresar un producto en el inventario.

Figura 53

Diagrama de secuencia crear producto



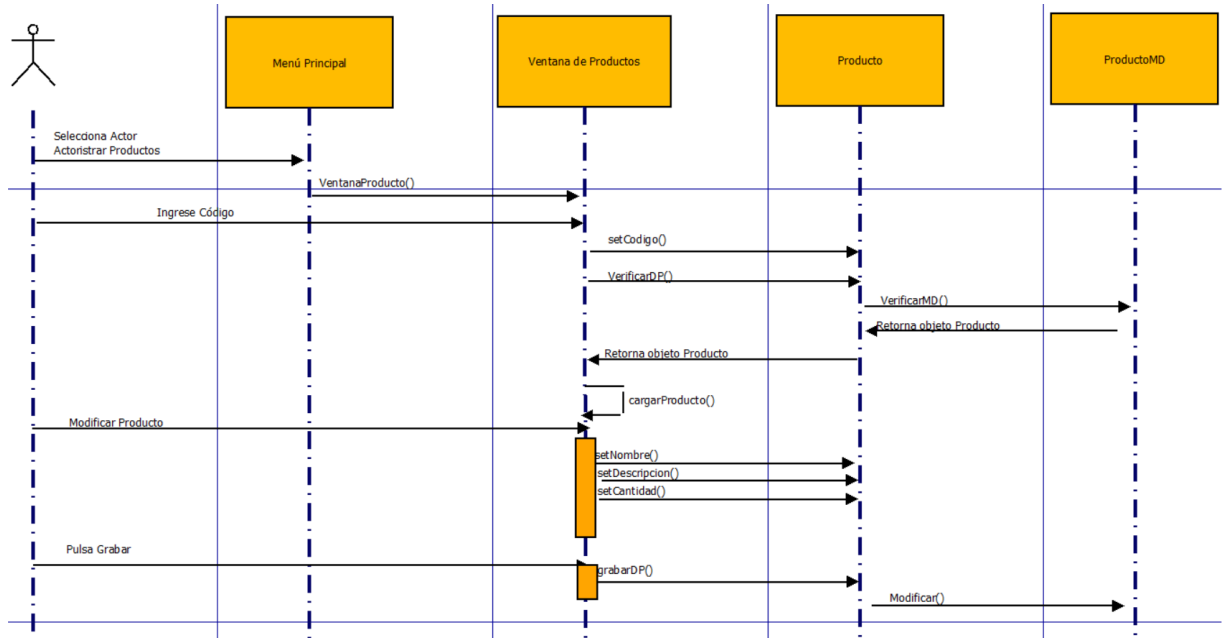
Autor: Cadena, S. (2023)

Modificar Producto:

El diagrama de secuencia del requerimiento funcional Gestionar Productos se centra en describir claramente los pasos que debe seguir el usuario para modificar un producto en el inventario.

Figura 54

Diagrama de secuencia modificar producto



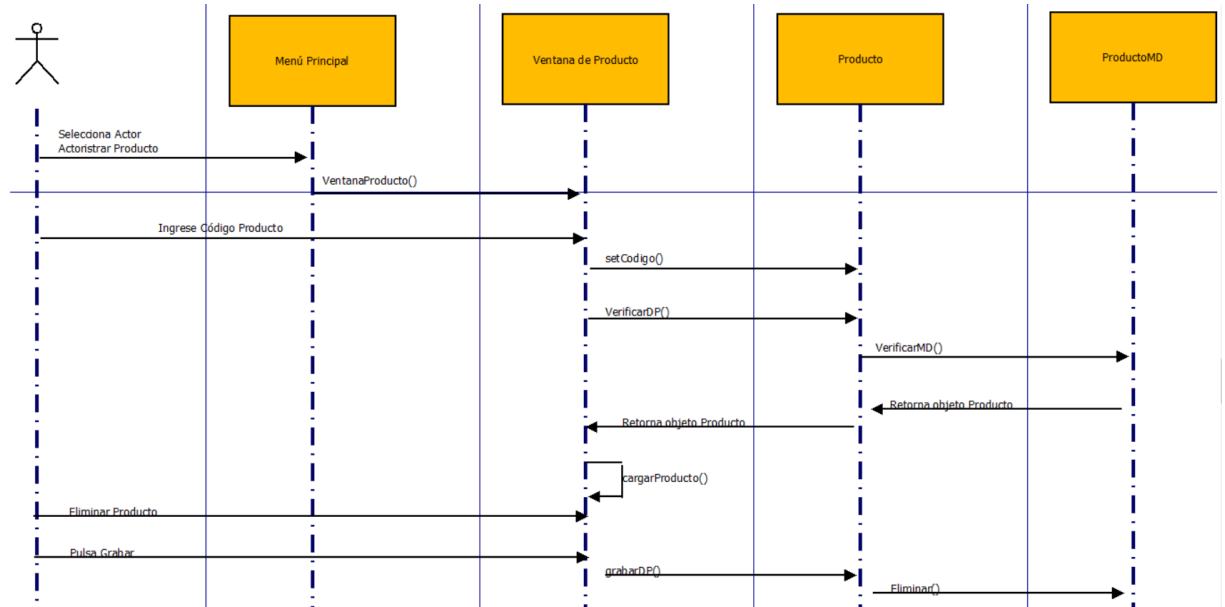
Autor: Cadena, S. (2023)

Eliminar Producto:

El diagrama de secuencia del requerimiento funcional Gestionar Productos se centra en describir claramente los pasos que debe seguir el usuario para eliminar un producto en el inventario.

Figura 55

Diagrama de secuencia eliminar producto



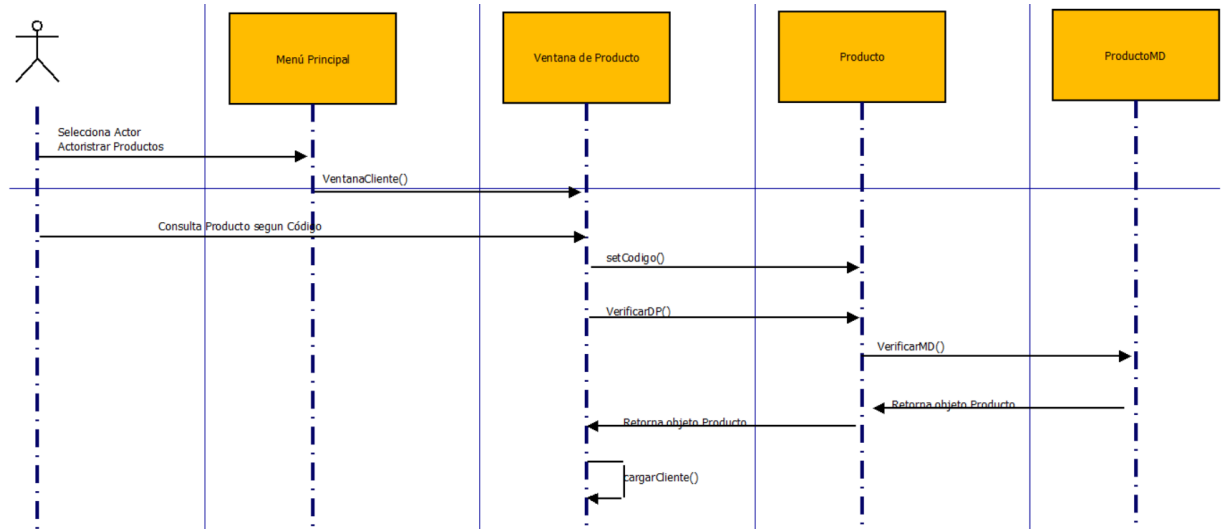
Autor: Cadena, S. (2023)

Buscar Producto por Código:

El diagrama de secuencia del requerimiento funcional Gestionar Productos se centra en describir claramente los pasos que debe seguir el usuario para buscar un producto en el inventario por código.

Figura 56

Diagrama de secuencia buscar producto por código



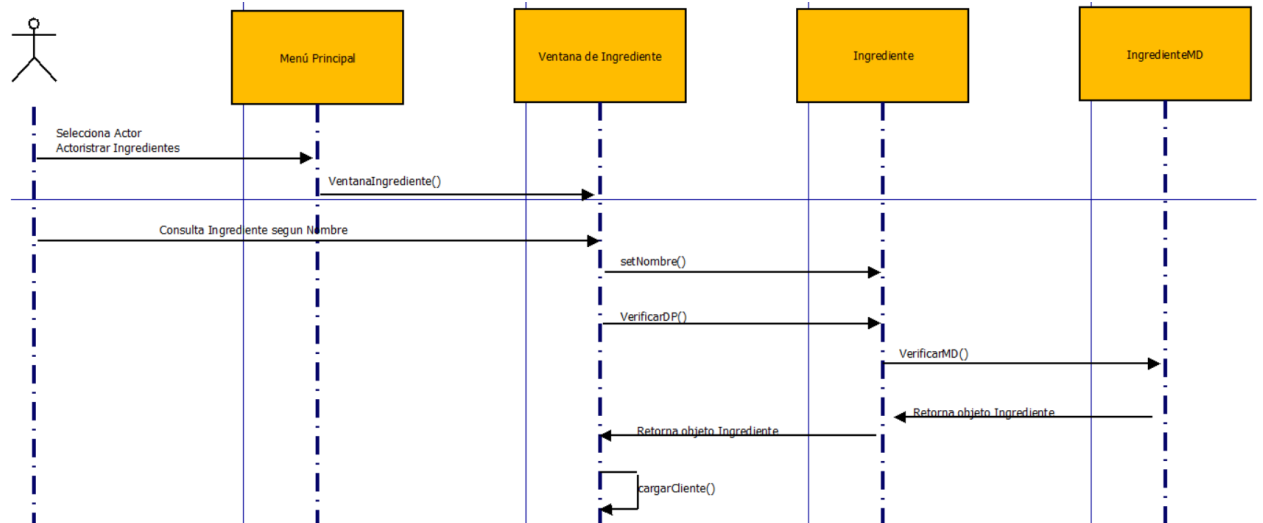
Autor: Cadena, S. (2023)

Buscar Producto por Nombre:

El diagrama de secuencia del requerimiento funcional Gestionar Productos se centra en describir claramente los pasos que debe seguir el usuario para buscar un producto en el inventario por nombre.

Figura 57

Diagrama de secuencia buscar producto por nombre



Autor: Cadena, S. (2023)

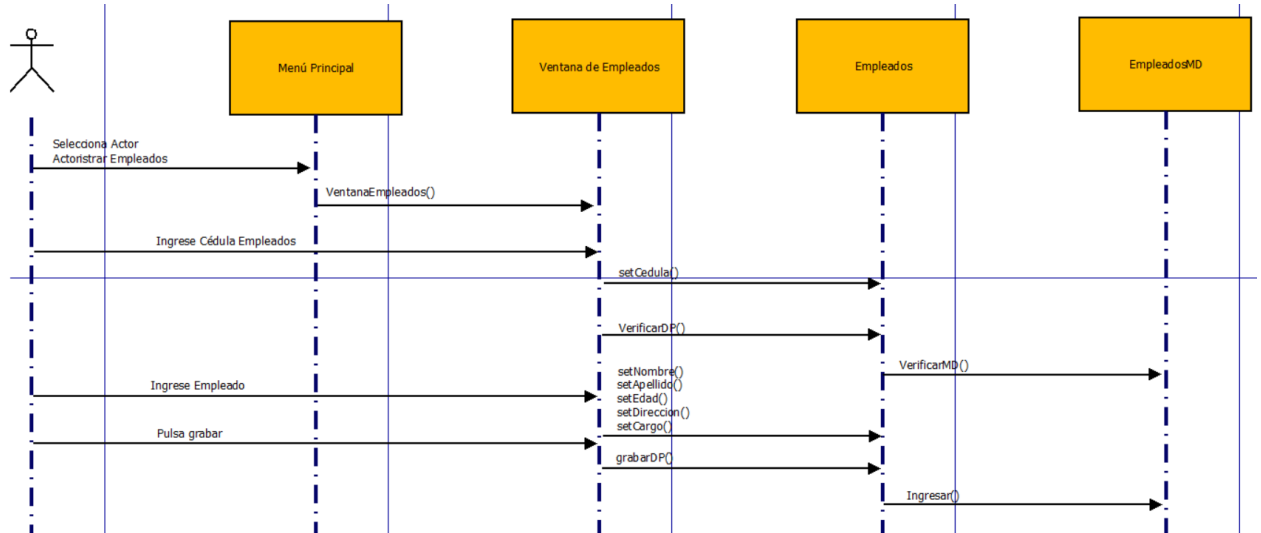
4.3.3.5. RF6 Gestionar Empleado

Crear Empleado:

El diagrama de secuencia del requerimiento funcional Gestionar Empleados se centra en describir claramente los pasos que debe seguir el usuario para crear un empleado del restaurante.

Figura 58

Diagrama de secuencia crear empleado



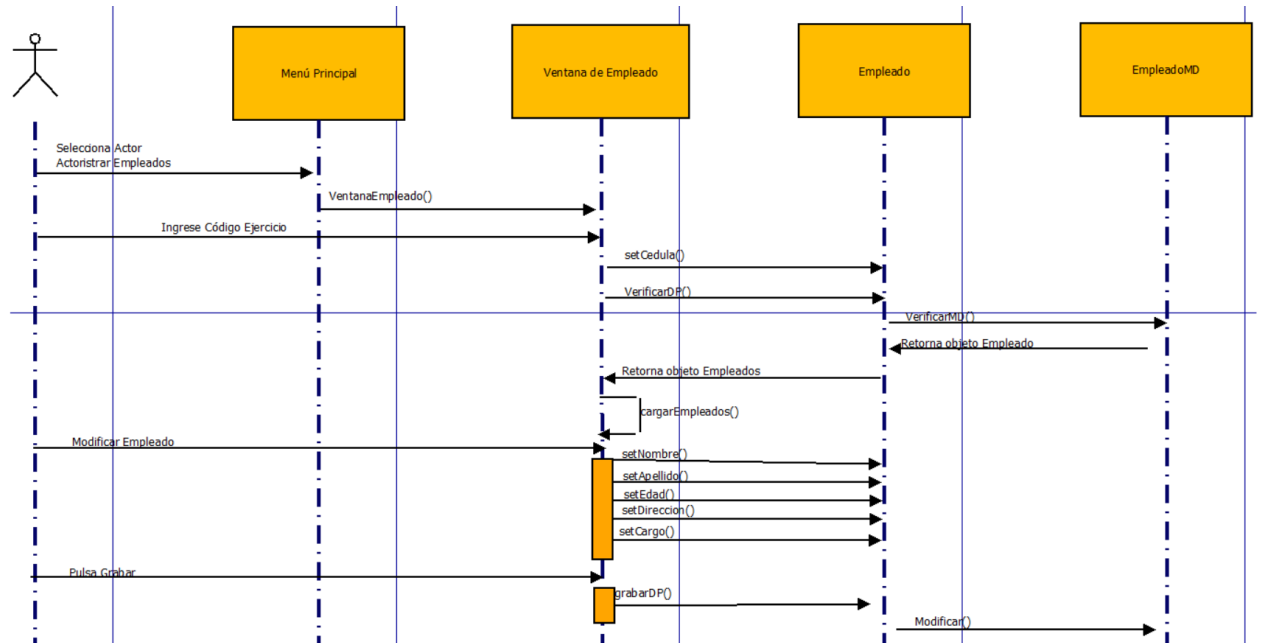
Autor: Cadena, S. (2023)

Modificar Empleado:

El diagrama de secuencia del requerimiento funcional Gestionar Empleados se centra en describir claramente los pasos que debe seguir el usuario para modificar un empleado del restaurante.

Figura 59

Diagrama de secuencia modificar empleado



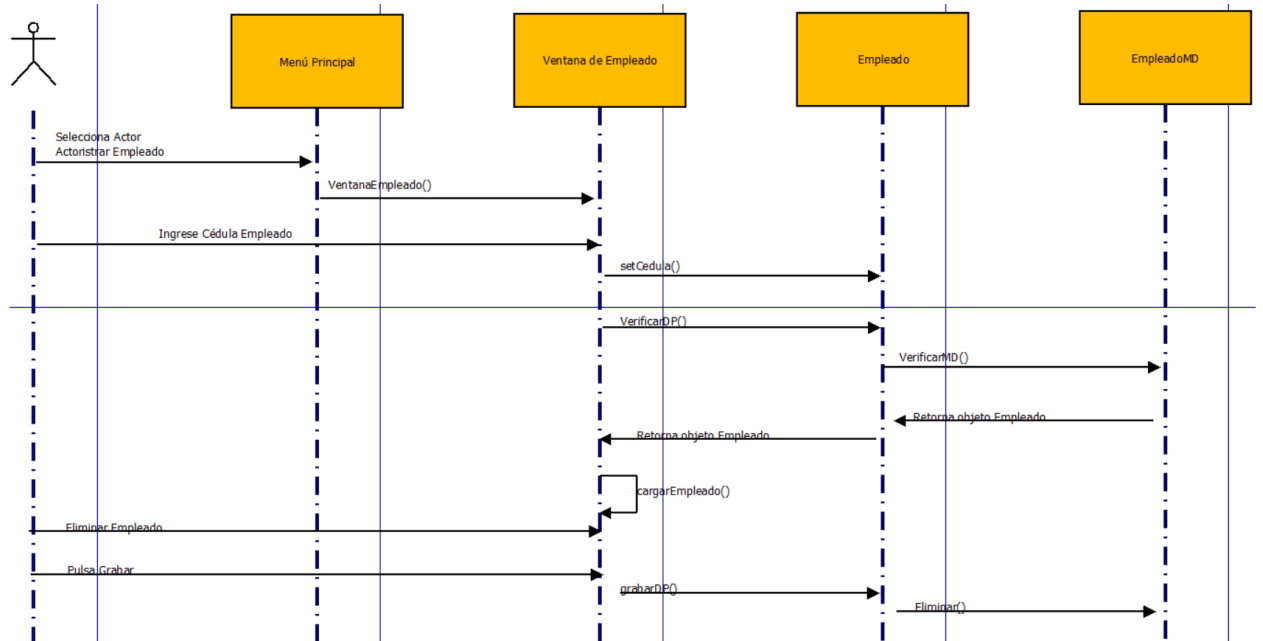
Autor: Cadena, S. (2023)

Eliminar Empleado:

El diagrama de secuencia del requerimiento funcional Gestionar Empleados se centra en describir claramente los pasos que debe seguir el usuario para eliminar un empleado del restaurante.

Figura 60

Diagrama de secuencia eliminar empleado



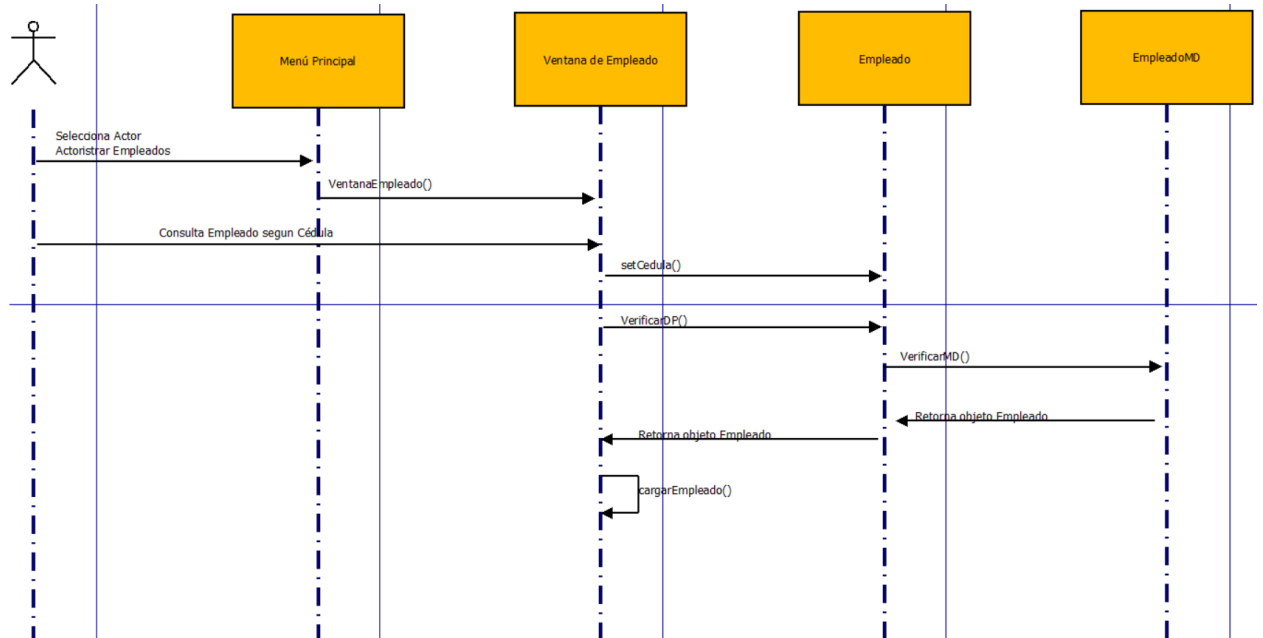
Autor: Cadena, S. (2023)

Buscar Empleado por Cédula:

El diagrama de secuencia del requerimiento funcional Gestionar Empleados se centra en describir claramente los pasos que debe seguir el usuario para buscar un empleado del restaurante por su cédula.

Figura 61

Diagrama de secuencia buscar empleado por cédula



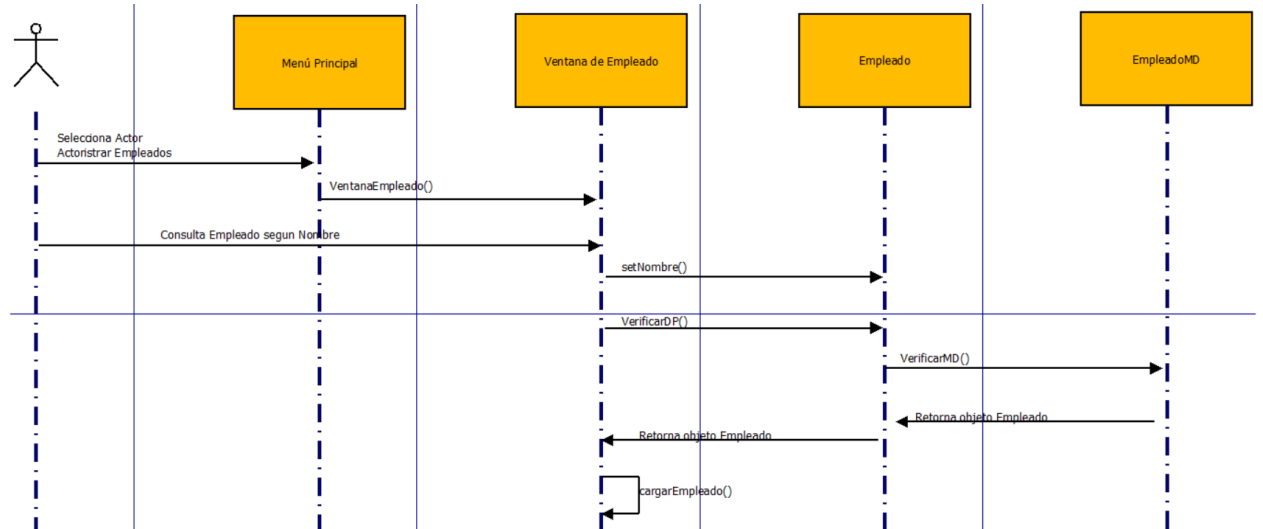
Autor: Cadena, S. (2023)

Buscar Empleado por Nombre:

El diagrama de secuencia del requerimiento funcional Gestionar Empleados se centra en describir claramente los pasos que debe seguir el usuario para buscar un empleado del restaurante por su nombre.

Figura 62

Diagrama de secuencia buscar empleado por nombre



Autor: Cadena, S. (2023)

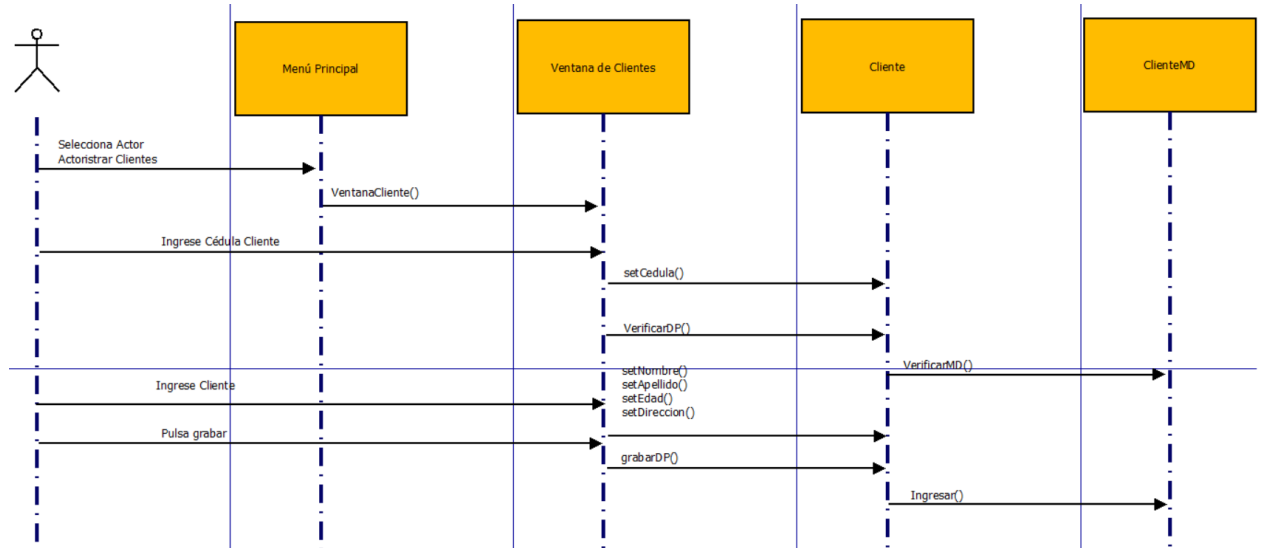
4.3.3.6. RF7 Gestionar Cliente

Crear Cliente:

El diagrama de secuencia del requerimiento funcional Gestionar Clientes se centra en describir claramente los pasos que debe seguir el usuario para crear un cliente del restaurante.

Figura 63

Diagrama de secuencia crear cliente



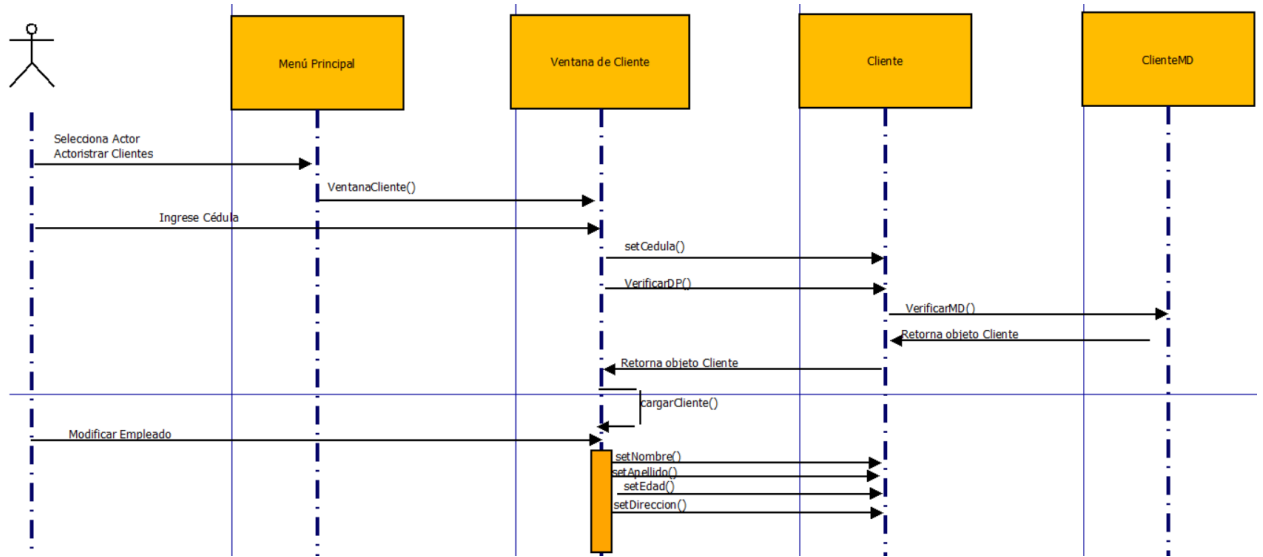
Autor: Cadena, S. (2023)

Modificar Cliente:

El diagrama de secuencia del requerimiento funcional Gestionar Clientes se centra en describir claramente los pasos que debe seguir el usuario para modificar un cliente del restaurante.

Figura 64

Diagrama de secuencia modificar cliente



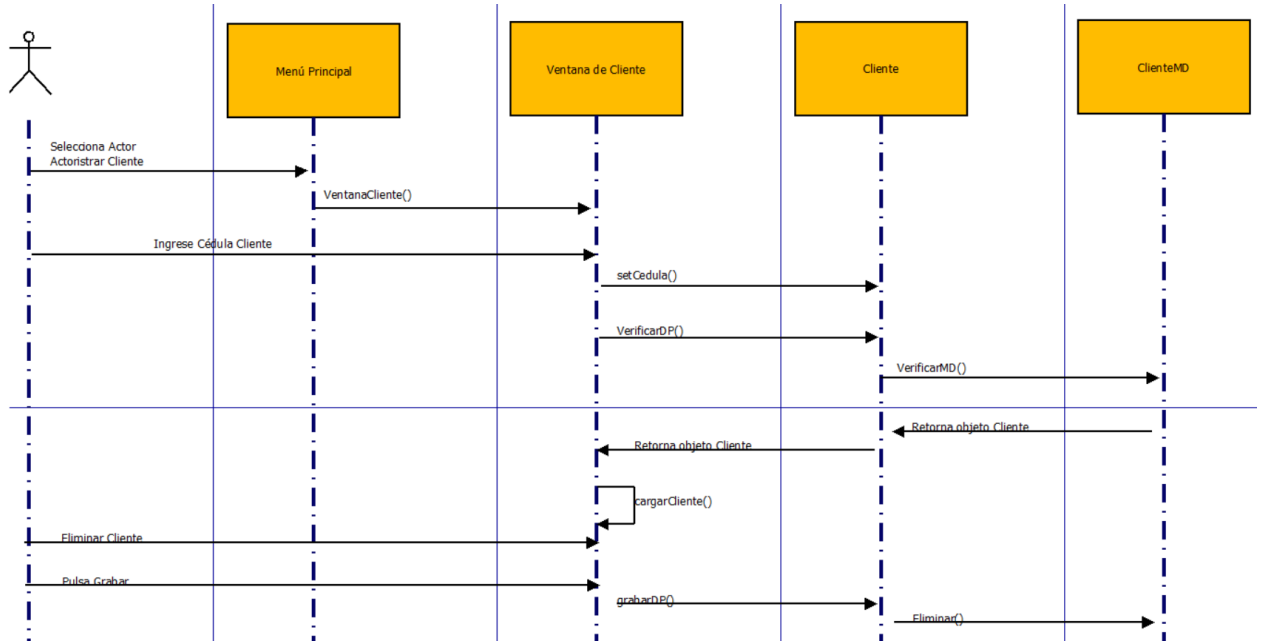
Autor: Cadena, S. (2023)

Eliminar Cliente:

El diagrama de secuencia del requerimiento funcional Gestionar Clientes se centra en describir claramente los pasos que debe seguir el usuario para eliminar un cliente del restaurante.

Figura 65

Diagrama de secuencia eliminar cliente



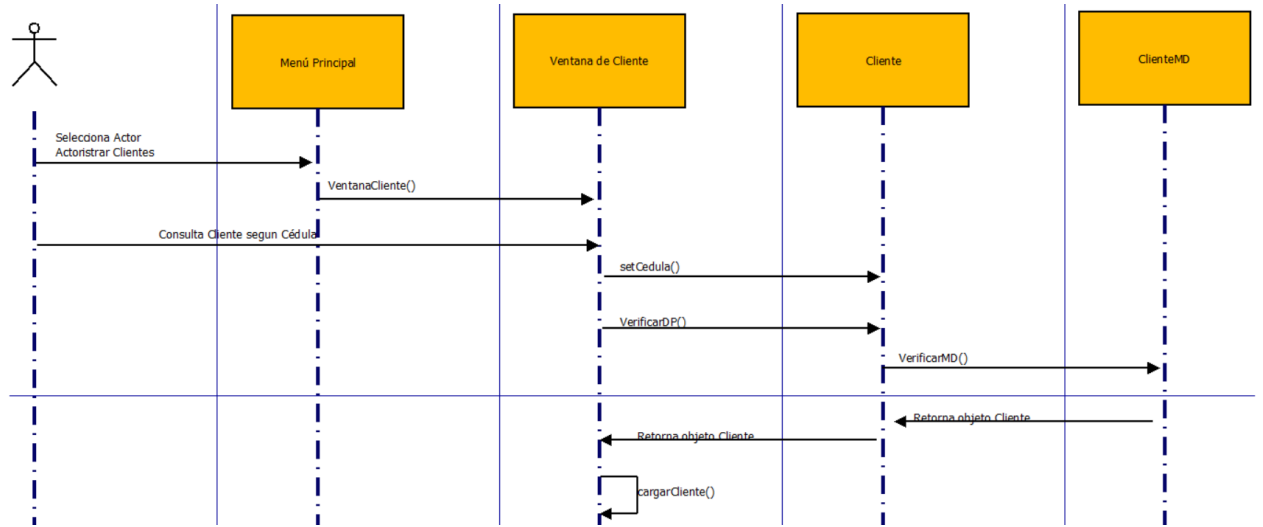
Autor: Cadena, S. (2023)

Buscar Cliente por Cédula:

El diagrama de secuencia del requerimiento funcional Gestionar Clientes se centra en describir claramente los pasos que debe seguir el usuario para buscar un cliente del restaurante por cédula.

Figura 66

Diagrama de secuencia buscar cliente por cédula



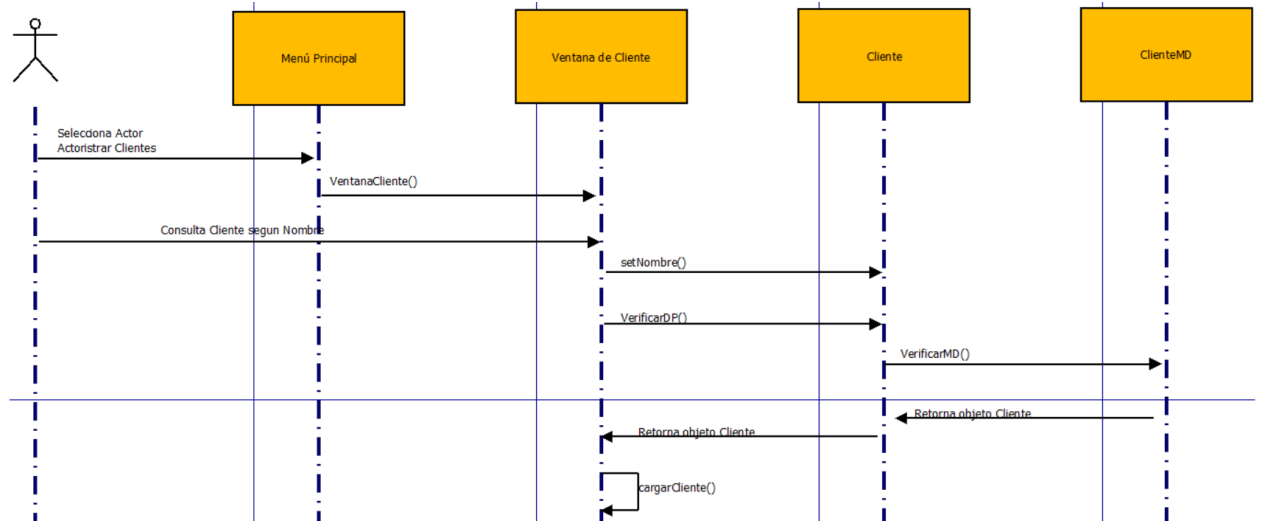
Autor: Cadena, S. (2023)

Buscar Cliente por Nombre:

El diagrama de secuencia del requerimiento funcional Gestionar Clientes se centra en describir claramente los pasos que debe seguir el usuario para buscar un cliente del restaurante por nombre.

Figura 67

Diagrama de secuencia buscar cliente por nombre



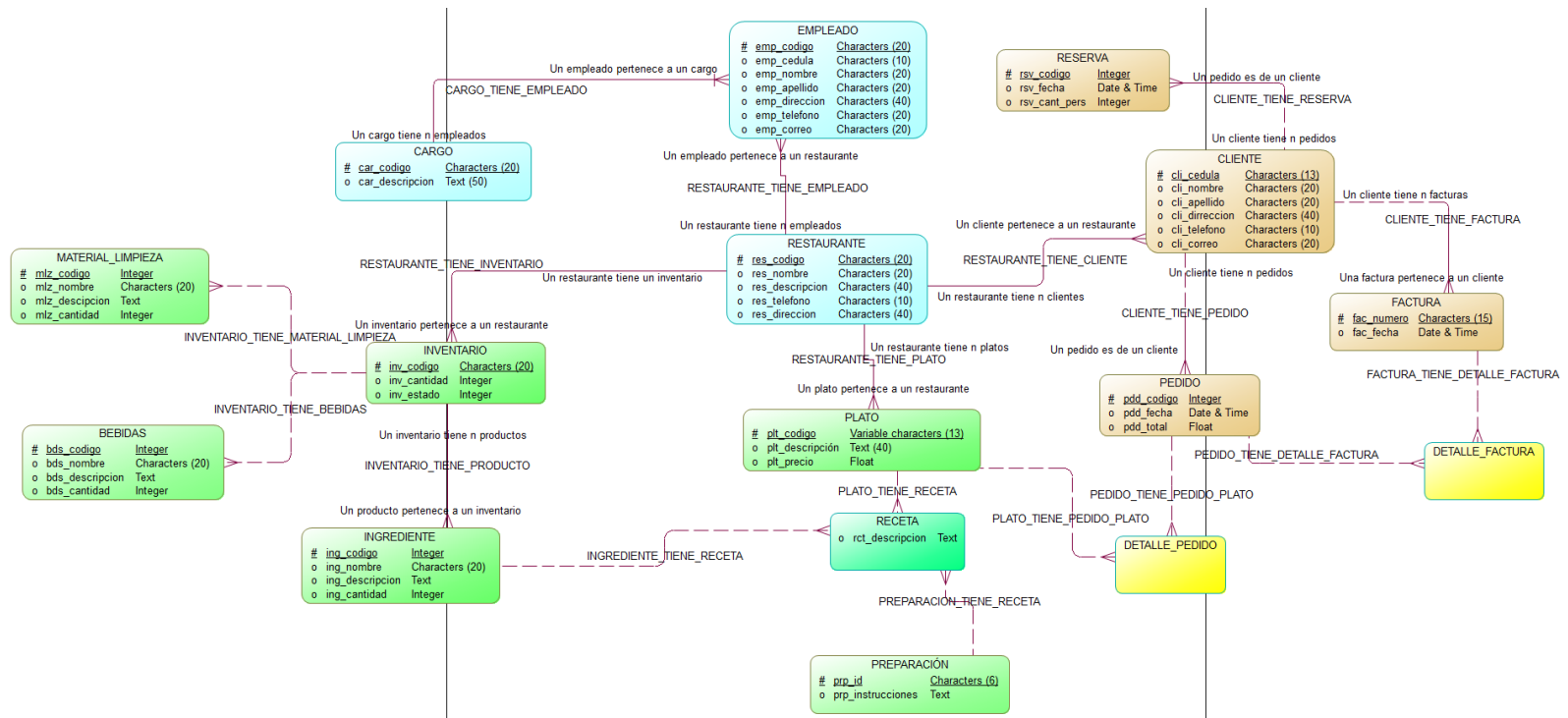
Autor: Cadena, S. (2023)

4.3.4. Diagrama de Clases

4.3.5. Diagrama Conceptual de Base de Datos

Figura 68

Diagrama conceptual del restaurante

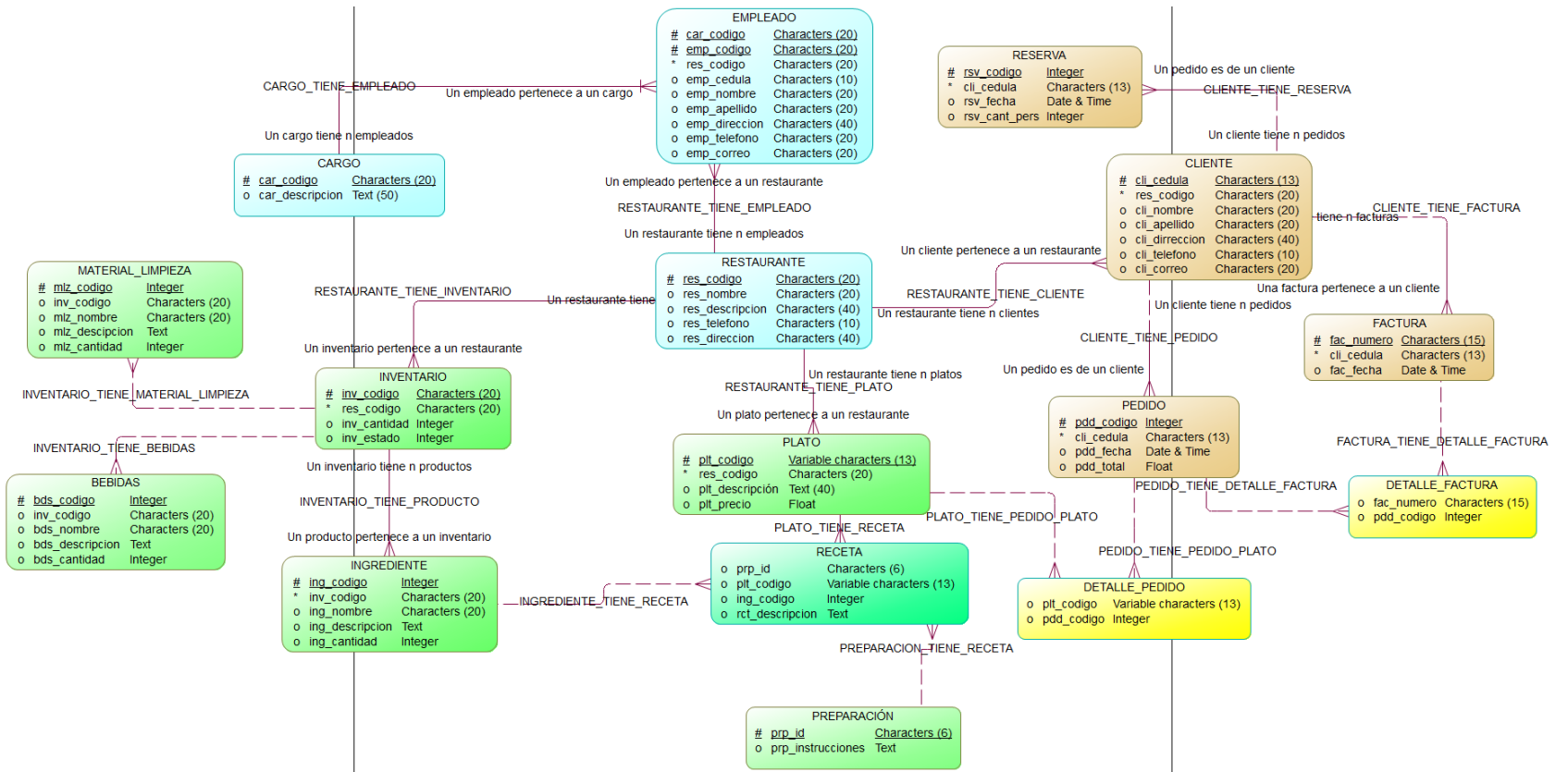


Autor: Cadena, S. (2023)

4.3.6. Diagrama Lógico de Base de Datos

Figura 69

Diagrama lógico del restaurante

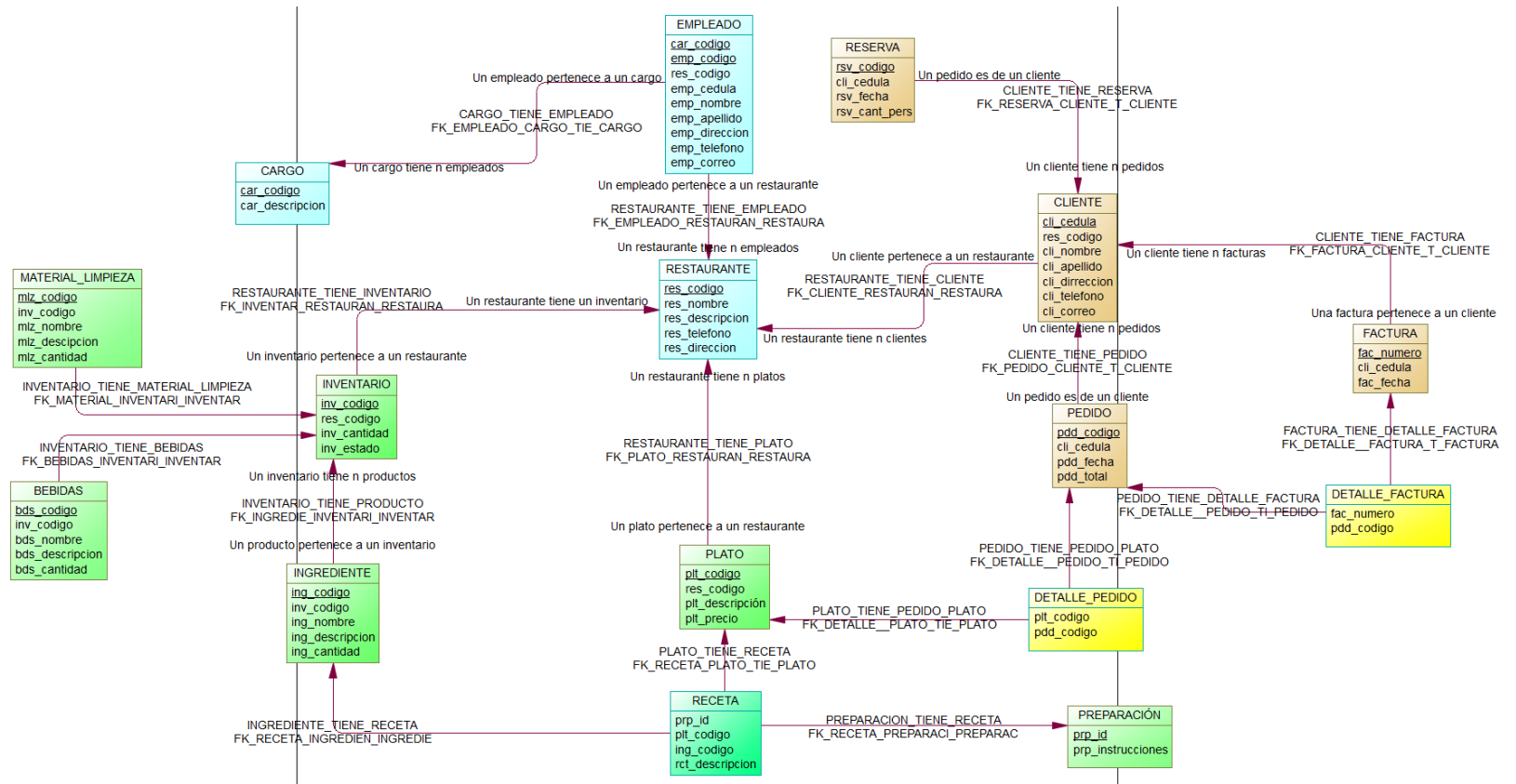


Autor: Cadena, S. (2023)

4.3.7. Diagrama Físico de Base de Datos

Figura 70

Diagrama físico del restaurante



Autor: Cadena, S. (2023)

4.3.8. Arquitectura de La Interfaz

Anexo B: Capturas de pantalla

CAPÍTULO V: IMPLEMENTACIÓN

5. Implementación de la aplicación

5.1. Implementación de la tesis

5.1.1. Productos Entregados

5.1.1.1. Diseño de la Base de Datos

El diseño de la base de datos consta de diferentes diagramas para la creación de las diferentes herramientas con las que cuenta el aplicativo web:

- Diagrama conceptual de base de datos.
- Diagrama lógico de base de datos.
- Diagrama físico de base de datos.
- Script de creación de base de datos.

5.1.1.2. Base de Datos

La base de datos con sus diferentes entidades para la gestión de la información del restaurante:

Tabla 17

Entidades

| Entidad | Descripción |
|-----------------|--|
| BEBIDAS | Contiene información de las bebidas del restaurante. |
| CARGO | Contiene información del cargo de los empleados del restaurante. |
| CLIENTE | Contiene información del cliente en el restaurante. |
| DETALLE FACTURA | Contiene información del detalle de factura del cliente. |
| DETALLE_PEDIDO | Contiene información del detalle del pedido del cliente. |
| EMPLEADO | Contiene información de los empleados en el restaurante. |
| FACTURA | Contiene información de factura de cada pedido. |

| | |
|-------------------|---|
| INGREDIENTE | Contiene información de los ingredientes del restaurante. |
| INVENTARIO | Contiene información de los materiales del restaurante. |
| MATERIAL LIMPIEZA | Contiene información de los materiales de limpieza del restaurante. |
| PEDIDO | Contiene información del pedido de cada cliente. |
| PLATO | Contiene información de los platos del restaurante. |
| PREPARACIÓN | Contiene información de las preparaciones del restaurante. |
| RECETA | Contiene información de las recetas del restaurante. |
| RESERVA | Contiene información de la reserva de cada cliente. |
| RESTAURANTE | Entidad en la que se registra información del restaurante. |

Autor: Cadena, S. (2023)

5.2. Sprint

5.2.1. Sprint 1

5.2.1.1. Sprint Planning

Según lo establecido en el sprint 1 se plantea establecer la conexión con la base de datos MySQL y continuar con la creación de los diferentes módulos de gestión del restaurante siendo los módulos de clientes y empleados, capaces de realizar funcionalidades de creación, modificación, eliminación y selección de los datos según el cargo del empleado.

5.2.1.2. Sprint Backlog

Tabla 18

Sprint Backlog

| Nombre | Descripción | Criterios para Aprobación |
|---|---|--|
| Establecer Conexión con la Base de Datos | Conexión con el motor de base de datos MySQL | Requiere de una conexión y funcionamiento adecuado |
| Página para Empleados | El usuario es capaz de crear, modificar, eliminar y visualizar los empleados del restaurante. | El usuario puede realizar las funcionalidades según su rol en el módulo de gestión de empleados. |
| Página para Clientes | El usuario es capaz de crear, modificar, eliminar y visualizar los empleados del restaurante. | El usuario puede realizar las funcionalidades según su rol en el módulo de gestión de clientes. |

Autor: Cadena, S. (2023)

5.2.1.3. Sprint Review

Al finalizar el Sprint 1 se realiza las pruebas para comprobar las funcionalidades de los diferentes módulos desarrollados se determinó que los módulos de empleados y clientes cumplen con los requerimientos para continuar el desarrollo del siguiente sprint.

5.2.1.4. Prueba de Sprint

Tabla 19

Prueba de Sprint

| Código | Nombre | Criterios para Aprobación | Estado |
|----------------|---|--|---------------|
| SP N°10 | Establecer Conexión con la Base de Datos. | Requiere de una conexión y funcionamiento adecuado | Aprobado |
| SP N°9 | Página para Empleados. | El usuario puede realizar las funcionalidades según su rol en el módulo de gestión de empleados. | Aprobado |
| SP N°6 | Página para Clientes. | El usuario puede realizar las funcionalidades según su rol en el módulo de gestión de clientes. | Aprobado |

Autor: Cadena, S. (2023)

5.2.2. Sprint 2

5.2.2.1. Sprint Planning

Según lo establecido en el sprint 2 se plantea continuar con la creación de los diferentes módulos de gestión del restaurante siendo los módulos de Plato e inventario, capaces de realizar funcionalidades de creación, modificación, eliminación y selección de los datos según el cargo del empleado.

5.2.2.2. Sprint Backlog

Tabla 20

Sprint Backlog

| Nombre | Descripción | Criterios para Aprobación |
|-------------------------------|--|---|
| Página para Platos | El usuario es capaz de crear, modificar, eliminar y visualizar los diferentes Platos del restaurante. | El usuario puede realizar las funcionalidades según su rol en el módulo de gestión de Platos. |
| Página para Inventario | El usuario es capaz de crear, modificar, eliminar y visualizar los Inventarios de productos del restaurante. | El usuario puede realizar las funcionalidades según su rol en el módulo de gestión de inventario. |

Autor: Cadena, S. (2023)

5.2.2.3. Sprint Review

Al finalizar el Sprint 2 se realiza las pruebas para comprobar las funcionalidades de los diferentes módulos desarrollados se determinó que los módulos de Plato e inventario cumplen con los requerimientos para continuar el desarrollo del siguiente sprint.

5.2.2.4. Prueba de Sprint

Tabla 21

Prueba de Sprint

| Código | Nombre | Criterios para Aprobación | Estado |
|---------------|----------------------|---|---------------|
| SP N°1 | Página para Plato | El usuario puede realizar las funcionalidades según su rol en el módulo de gestión de Platos. | Aprobado |
| SP N°5 | Página para Producto | El usuario puede realizar las funcionalidades según su rol en el módulo de gestión de Producto. | Aprobado |

Autor: Cadena, S. (2023)

5.2.3. Sprint 3

5.2.3.1. Sprint Planning

Según lo establecido en el sprint 3 se plantea continuar con la creación de los diferentes módulos de gestión del restaurante siendo los módulos de reservas y pedidos, capaces de realizar funcionalidades de creación, modificación, eliminación y selección de los datos según el cargo del empleado.

5.2.3.2. Sprint Backlog

Tabla 22

Sprint Backlog

| Nombre | Descripción | Criterios para Aprobación |
|-----------------------------|---|---|
| Página para Reservas | El usuario es capaz de crear, modificar, eliminar y visualizar los diferentes Reservas de clientes del restaurante. | El usuario puede realizar las funcionalidades según su rol en el módulo de gestión de reservas. |
| Página para Pedidos | El usuario es capaz de crear, modificar, eliminar y visualizar los Pedidos de clientes del restaurante. | El usuario puede realizar las funcionalidades según su rol en el módulo de gestión de pedidos. |

Autor: Cadena, S. (2023)

5.2.3.3. Sprint Review

Al finalizar el Sprint 3 se realiza las pruebas para comprobar las funcionalidades de los diferentes módulos desarrollados se determinó que los módulos de reservas y pedidos cumplen con los requerimientos.

5.2.3.4. Prueba de Sprint

Tabla 23

Prueba de Sprint

| Código | Nombre | Criterios para Aprobación | Estado |
|---------------|----------------------|---|---------------|
| SP N°3 | Página para Reservas | El usuario puede realizar las funcionalidades según su rol en el módulo de gestión de reservas. | Aprobado |
| SP N°2 | Página para Pedidos | El usuario puede realizar las funcionalidades según su rol en el módulo de gestión de pedidos. | Aprobado |

Autor: Cadena, S. (2023)

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Durante el desarrollo de este proyecto de investigación, se logró diseñar y desarrollar una aplicación web responsiva para la gestión de reservas y administración del restaurante. Esta aplicación ofrece una solución eficiente y moderna para simplificar y mejorar la gestión de reservas y el control administrativo en restaurantes.
- Se ha demostrado la importancia de utilizar tecnologías web como HTML, CSS, Javascript y PHP, que proporcionan las herramientas necesarias para crear interfaces atractivas y funcionales, y ofrecen una experiencia de usuario intuitiva y fácil de usar.
- La metodología de desarrollo de software utilizada, en particular SCRUM, demostró ser altamente efectiva para el desarrollo de proyectos ágiles y colaborativos. La implementación de sprints permitió una entrega incremental de funcionalidades, lo que facilitó la adaptación a los cambios y la obtención de resultados rápidos.
- La correcta identificación de roles y requerimientos fue fundamental para el éxito del proyecto. Al asignar roles claros y definidos, se pudo establecer responsabilidades y asegurar una comunicación efectiva entre los miembros del equipo. Además, la identificación de requerimientos, tanto funcionales como no funcionales, permitió desarrollar una aplicación que cumple con las expectativas y necesidades de los usuarios.

Recomendaciones

- Se recomienda realizar pruebas exhaustivas de la aplicación antes de su implementación completa. Esto asegurará que todos los aspectos funcionales y de rendimiento se encuentren en óptimas condiciones, evitando posibles problemas en un futuro y lograr mejorar la experiencia del usuario.
- Es importante que el personal que realice mantenimiento de la aplicación siga actualizándose en cuanto a las nuevas tecnologías y tendencias en el desarrollo de aplicaciones web. Esto permitirá aprovechar las ventajas de las herramientas más recientes y mantener la aplicación al día con los avances tecnológicos.
- Se sugiere mantener un proceso de retroalimentación constante con los usuarios finales de la aplicación. Esto permitirá recibir comentarios y sugerencias para mejorar continuamente la aplicación y adaptarla a las necesidades cambiantes de los restaurantes y sus clientes.
- Considerar la posibilidad de implementar medidas de seguridad adicionales en la aplicación, como la encriptación de datos sensibles y la implementación de autenticación y autorización seguras. Esto garantizará la protección de la información y la confidencialidad de los datos de los usuarios y del restaurante.
- Ante cualquier cambio en la aplicación se recomienda documentar los manuales de usuario y cualquier otra documentación relevante. Esto facilitará el mantenimiento y la comprensión de la aplicación en el futuro, tanto para el equipo de desarrollo como para posibles colaboradores o administradores posteriores.

- Finalmente, se sugiere evaluar la posibilidad de expandir la funcionalidad de la aplicación en el futuro, considerando nuevas características y servicios que puedan agregar valor a los restaurantes y a los usuarios finales. Esto permitirá mantener la aplicación competitiva y adaptada a las necesidades en constante evolución del sector de la gestión de reservas y administración de restaurantes.

BIBLIOGRAFÍA

- Al-Daraiseh, A., & Samarah, H. (2019). Agile Software Development: An Empirical Study of Challenges and Benefits. *Journal of Computer Information Systems*, 234-243.
- Ali, H., & Shallal, M. (2019). A Systematic Review on the Application of UML to Model Software Systems. *Journal of Engineering*.
- Almagro, C. A. (Diciembre de 2011). *Universidad de Granada*. Obtenido de Departamento de Lenguajes y Sistemas Informáticos: <https://lsi.ugr.es/curena/doce/lp/tr-11-12/lp-c01-impr.pdf>
- Álvarez, J. (29 de septiembre de 2017). *Entorno de Desarrollo intencional basado en XML*. [Tesis de Pregrado], Repositorio de la Universidad Politécnica de Madrid. Obtenido de https://oa.upm.es/49793/1/PFC_JOSE_ANTONIO_ALVAREZ_PEREZ.pdf
- Arino, J. (2021). ¿Qué es la metodología Scrum? *Scrum Master*. Obtenido de <https://metodologiascrum.top/que-es-la-metodologia-scrum/>
- Azeem, M., & Hussain, S. (2020). Software Development Methodologies: A Review. *Journal of Software Engineering and Applications*, 647-670.
- Alonso-Aranda, C. (2019). *MODELO-VISTA-CONTROLADOR. LENGUAJE UML*. <http://tauja.ujaen.es/jspui/handle/10953.1/11437>
- Lozano Banqueri, J. M. (2018). *Creación y gestión de una base de datos con MySQL y phpMyAdmin*. <http://tauja.ujaen.es/jspui/handle/10953.1/9445>
- Astigarraga, J., & Cruz-Alonso, V. (2022). ¡Se puede entender cómo funcionan Git y GitHub! *Ecosistemas*, 31(1), Article 1. <https://doi.org/10.7818/ECOS.2332>
- Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice* (3 ed.). Addison-Wesley.
- Bhagwat, R., & Patil, S. (2021). A Comparative Study of Relational Database Management Systems. *International Journal of Computer Science and Information Security*.
- Briones Chauca, K. F. (2019). *Desarrollo de una página web para automatizar el sistema de ventas y soporte técnico mediante chat en tiempo real para la Empresa Faceti S.A. de la ciudad de Guayaquil*. [Thesis, Universidad de Guayaquil. Facultad de Ingeniería Industrial. Carrera de Licenciatura en Sistemas de Información.]. <http://repositorio.ug.edu.ec/handle/redug/41911>
- Castañeda Rodríguez, L. D. (2016). *Implementación de JQuery Mobile para la creación de sitios y/o aplicaciones Web intuitivas, amigables y responsivas* [BachelorThesis,

- <http://repositorio.upsin.edu.mx>].
- <http://repositorio.upsin.edu.mx/formatos/LuisDavidFinal2535.pdf>
- C. Xia, G. Y. (2009). Efficient Implement of ORM (Object/Relational Mapping) Use in J2EE Framework: Hibernate. *Efficient Implement of ORM (Object/Relational Mapping) Use in J2EE Framework: Hibernate*. Wuhan, Hubei, China: International Conference on Computational Intelligence and Software Engineering.
- Casas, J. (2021). *Diseño conceptual de bases de datos*. Universitat Oberta de Catalunya. Obtenido de http://cv.uoc.edu/annotation/cb826b689abc472d8fb5b2519840058b/699689/PID_00213704/PID_00213704.html
- Coronel , C., & Morris, S. (2018). *Sistemas de bases de datos: diseño, implementación y gestión* (13 ed.).
- Coronel, C., & Morris, S. (2016). *Database Systems: Design, Implementation, and Management* (12 ed.). Cengage Learning.
- Espinosa-Hurtado, R. (2021). Análisis comparativo para la evaluación de frameworks usados en el desarrollo de aplicaciones web. *CEDAMAZ*, 11(2), Article 2. <https://doi.org/10.54753/cedamaz.v11i2.1182>
- Eduardo Polo Ortega, F. J. (2015). *Servicios de red e Internet*. Madrid, España: RA-MA Editorial.
- Elmasri, R., & Navathe, S. (2019). *Relational Databases: Concepts, Design, and Implementation* (3 ed.). Pearson.
- Fowler, M. (2019). *UML destilado: una breve guía del lenguaje estándar de modelado de objetos* (4 ed.). Addison-Wesley Professional.
- Flanagan, D. (2020). *JavaScript: la guía definitiva*. O'Reilly Media.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- García-Molina, H., Ullman, J., & Widom, J. (2021). *Database Systems: The Complete Book* (3 ed.). Pearson.
- GilJin Yang, B. C. (01 de 2014). *Research Gate*. Obtenido de www.researchgate.net/publication/298642533_Implementation_of_HTTP_live_streaming_for_an_IP_camera_using_an_open_source_multimedia_converter

- Graydon, M. &. (7 de August de 2019). *'Connecting the unconnected': a critical assessment of US satellite Internet services*. Obtenido de SAGE JOURNALS: <https://doi.org/10.1177/0163443719861835>
- Grimm , N., & Hans, D. (2018). Applied Research: Historical Background and Current Trends. *Research Quarterly for Exercise and Sport*, 121-132.
- Guniganti, R. &. (2013). A Comparison of RTMP and HTTP Protocols with respect to Packet Loss and Delay Variation based on QoE. *semanticscholar.org*.
- Gupta, A., & Chauhan, S. (2015). Database Management Systems: A Review. *Journal of Advanced Research in Computer Science and Software Engineering*, 329-334.
- Gutiérrez, J. J. (12 de 05 de 2014). *Qué es un framework web*. Obtenido de http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework
- Gómez García, D. E. (2018). *Desarrollo del sistema de requisiciones para la empresa hidroeléctrica Abanico S.A. Aplicando el entorno de programación Node.js* [BachelorThesis, Escuela Superior Politécnica de Chimborazo]. <http://dspace.epoch.edu.ec/handle/123456789/9110>
- Grau, X. F., & Segura, M. I. S. (2000). *Desarrollo Orientado a Objetos con UML*.
- Guapi Auquilla, M. J. (2018). *Diseño metodológico para el desarrollo de interfaces gráficas en páginas web utilizando los lenguajes HTML 5 y CSS 3* [BachelorThesis, Riobamba]. <http://dspace.unach.edu.ec/handle/51000/5163>
- Hernández , R., Fernández , C., & Baptista , P. (2014). *Metodología de la investigación: Enfoque cualitativo y cuantitativo* (Vol. 6th). McGraw-Hill.
- Highsmith, J. (2019). *Agile Project Management: Creating Innovative Products*. Addison-Wesley Professional.
- Ignacio, C., & Paola, V. (2015). *Metodologías actuales de desarrollo de software. 2015*. https://www.ecorfan.org/bolivia/researchjournals/Tecnologia_e_innovacion/vol2num5/Tecnologia_e_Innovacion_Vol2_Num5_6.pdf
- I. Fette, A. M. (December de 2011). *Internet Engineering Task Force (IETF)* . Obtenido de <https://www.hjp.at/doc/rfc/rfc6455.html>
- IEEE Computer Society. (2020). *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*. Obtenido de <https://ieeexplore.ieee.org/document/9136251>
- Instituto Nacional de Estadísticas y Censos. (2019). *www.ecuadorencifras.gob.ec*. Obtenido de <https://www.ecuadorencifras.gob.ec/documentos/web->

inec/Estadísticas_Sociales/TIC/2019/201912_Principales_resultados_Multiproposito_TIC.pdf

- JetBrains. (2023). *¿Qué es un IDE?* Obtenido de <https://www.jetbrains.com/idea/ide/>
- Joskowicz, J. (10 de 02 de 2008). Reglas y prácticas en eXtreme Programming. Universidad de Vigo, 22. Obtenido de https://d1wqtxts1xzle7.cloudfront.net/31398587/xp_-_jose_joskowicz-with-cover-page-v2.pdf?Expires=1625441610&Signature=HZfoDu6RCpoB-dKMMuLRNmZRaiz0cWUrcjbndtRyjECrK33QWDAGtINDg1Cnw9kvQJ9Psul9gXX8CRJculpl5KBhJgUMc-blqu72mdVo6cpvqyy3-XejGZUvukkePQRHmxxpb-Ddq
- PHP. (2023). *Manual de PHP*. Obtenido de Sitio oficial de PHP: <https://www.php.net/manual/es/>
- Kumar, R. (2019). *Research Methodology: A Step-by-Step Guide for Beginners*. SAGE Publications.
- Kruchten, P. (2015). *Planos Arquitectónicos: El Modelo de "4+1" Vistas de la Arquitectura del Software*.
- Lagatree, K. (2006). Keep it Together. En K. Lagatree, *Keep It Together: 200+ Tips, Tricks, Lists, and Solution for EverydayLife* (pág. 432). Random House Reference.
- Ley De Comercio Electrónico, Ley 67 (Congreso Nacional 17 de 05 de 2002).
- Ley Orgánica De Comunicación, 22 (Legislativo 25 de 06 de 2013).
- Llontop Espinoza, A. C. (2019). *Sistema web basado en el Framework Codeigniter para el proceso de control de incidencias en la Municipalidad de Breña* [Universidad César Vallejo]. <https://repositorio.ucv.edu.pe/handle/20.500.12692/47264>
- López, D. A., Burgos-Salazar, S. P., Solórzano-Solórzano, S. E., & Mejía-Condolo, M. F. (2020). *La economía de las empresas del Ecuador en el contexto de la pandemia The Ecuadorian business economy in the context of the pandemic Economía empresarial do Equador no contexto da pandemia*. 5(1).
- López Herrera, P. (2016, March 18). *Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL - CORE*. https://core.ac.uk/display/80528621?utm_source=pdf&utm_medium=banner&utm_campaign=pdf-decoration-v1
- Lozada, J. (2014). Investigación Aplicada: Definición, Propiedad Intelectual e Industria. *CienciAmérica: Revista de divulgación científica de la Universidad Tecnológica Indoamérica*, 3(1), 47–50.

- Lozano Banqueri, J. M. (2018). *Creación y gestión de una base de datos con MySQL y phpMyAdmin*. <http://tauja.ujaen.es/jspui/handle/10953.1/9445>
- Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software*. <https://repositorio.uca.edu.ar/handle/123456789/522>
- Martínez, C. (2019, January 1). *Implantación de metodologías ágiles en un equipo de desarrollo de software—CORE*. *Implantación de Metodologías Ágiles En Un Equipo de Desarrollo de Software*. <https://core.ac.uk/works/46501116>
- Montero, B. M., Cevallos, H. V., & Cuesta, J. D. (2018). *Agile methodologies against traditional methods in the software development process*. <https://gc.scalahed.com/recursos/files/r161r/w25597w/438760423-269-823-1-PB-pdf.pdf>
- MDN Web Docs. (2023). *CSS - Hojas de estilo en cascada*. Obtenido de <https://developer.mozilla.org/en-US/docs/Web/CSS>
- Microsoft. (2023). *Documentación del IDE de Visual Studio*. Obtenido de <https://learn.microsoft.com/en-us/visualstudio/ide/?view=vs-2022>
- Microsoft. (30 de junio de 2022). *MVC (Model-View-Controller) Architectural Pattern*. Obtenido de <https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/>
- Mozilla MDN Web Docs. (2023). *JavaScript*. Obtenido de <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Muñoz, J. (25 de 07 de 2006). *maestrosdelweb*. Obtenido de www.maestrosdelweb.com/intersatelite/
- MySQL. (2023). *Sitio oficial de MySQL*. Obtenido de <https://www.mysql.com/>
- Oracle Corporation and/or its affiliate. (2020). *What is MySQL*. Obtenido de <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- Pérez Ibarra, S. G. (17 de 06 de 2021). *Red de Universidades con Carreras en Informática*. Obtenido de <http://sedici.unlp.edu.ar/handle/10915/120476>
- Pérez, M. (2011). *Microsoft SQL Server 2008 R2. Motor de base de datos y administración*. RC Libros.
- Pressman, R., & Maxim, B. (2020). *Software Engineering: A Practitioner's Approach*. McGraw-Hill.
- Pressman, S., & Maxim, B. (2020). *Software Engineering: A Practitioner's Approach* (9 ed.). McGraw-Hill.

- Quito Turismo. (2020). *Quito en cifras 2020_ turismo y servicios*. <http://quito-turismo.gob.ec/estadisticas/>
- Ramírez, M. R., Rojas, E. M., & Cisneros, F. R. (2018). *Metodología SCRUM y desarrollo de Repositorio Digital*.
https://media.proquest.com/media/hms/PFT/1/eJDZ8?_s=ls80spXmqPbwssplWafUZQRB8%2B8%3D
- Ramakrishnan, R., & Gehrke, J. (2020). *Database Management Systems* (4 ed.). McGraw-Hill.
- Real Academia Española. (2021). REAL ACADEMIA ESPAÑOLA.
- Red Hat. (20 de enero de 2023). *¿Qué es y para qué sirve un IDE?* Obtenido de <https://www.redhat.com/es/topics/middleware/what-is-ide>
- Richard, A. (2017). Can Weather Affect Satellite Internet? *Hearst Newspapers*, 1. Obtenido de Can Weather Affect Satellite Internet?: <http://smallbusiness.chron.com/can-weather-affect-satellite-internet-26822.html>
- Salkind, N. (2010). *Applied Research*. En Encyclopedia of Research Design.
- Sampieri, R. F. (2014). Definiciones de los enfoques cuantitativo y cualitativo, sus similitudes y diferencias. En C. F. Roberto Hernández Sampieri. RH Sampieri, Metodología de la Investigación.
- Schwaber, K. (2004). *Agile Project Management with Scrum*. Microsoft Press.
- Singh, G., & Chauhan, S. (2021). A Survey on Database Management Systems. *Journal of Computer Science and Mobile Computing*, 10-17.
- The PHP Group. (2001-2020). *php.net*. Obtenido de <https://www.php.net/manual/es/intro-what-is.php>
- Sutherland, J., & Schwaber, K. (2017). An Overview of Scrum Methodology in Agile Development. *International Journal of Advanced Research in Computer Science*.
- Taylor Otwell. (2020). *Laravel*. Obtenido de <https://laravel.com/>
- The PHP Group. (2001-2020). *php.net*. Obtenido de <https://www.php.net/manual/es/intro-what-is.php>
- Velásquez, S. M., Montoya, J. D. V., Adasme, M. E. G., Zapata, E. J. R., Pino, A. A., & Marín, S. L. (2019). Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software. *Revista CINTEX*, 24(2), Article 2. <https://doi.org/10.33131/24222208.334>
- Vijayasarithi, P., & Dhandapani, A. (2019). Agile vs Traditional Methodologies: A Comparative Study. *International Journal of Pure and Applied Mathematics*, 477-485.

W3C. (2013). *Introducción al CSS*. Obtenido de <https://www.w3.org/Style/CSS/Overview.en.html>

GLOSARIO DE TÉRMINOS

- **Página Web:** Un documento electrónico que se muestra en un navegador web y que contiene información como texto, imágenes y videos.
- **HTML (HyperText Markup Language):** Lenguaje de marcado utilizado para estructurar y presentar el contenido de una página web.
- **CSS (Cascading Style Sheets):** Lenguaje utilizado para definir la apariencia y el formato visual de una página web.
- **Javascript:** Lenguaje de programación utilizado para crear interactividad y funcionalidades en una página web.
- **PHP:** Lenguaje de programación utilizado como diseñado para el desarrollo web.
- **Arquitectura de Software:** Estructura y organización de un sistema de software, incluyendo la interacción entre sus componentes.
- **Metodología Aplicada:** Enfoque sistemático para desarrollar conocimiento práctico y aplicado en un área específica.
- **Metodología para el Desarrollo de Software:** Proceso y conjunto de pasos a seguir para idear, implementar y mantener un producto de software.
- **UML (Unified Modeling Language):** Lenguaje de modelado utilizado para visualizar, especificar y documentar los aspectos de un sistema de software.
- **Base de Datos:** Conjunto organizado de datos relacionados, almacenados de forma estructurada para facilitar su gestión y consulta.
- **Metodologías Ágiles de Desarrollo:** Enfoques flexibles y colaborativos para el desarrollo de software, que priorizan la adaptación al cambio y la entrega incremental.

- **SCRUM:** Metodología ágil para la gestión de proyectos que se centra en la colaboración, la adaptabilidad y la entrega de resultados rápidos.
- **Sprint:** Iteración de trabajo en el marco de la metodología SCRUM, generalmente de corta duración, durante la cual se desarrollan y entregan funcionalidades específicas.

ANEXOS

Anexo A: Script

```
/*=====*/
```

```
/* DBMS name:   MySQL 5.0           */
```

```
/* Created on:  1/6/2023 17:45:33   */
```

```
/*=====*/
```

```
alter table BEBIDAS
```

```
drop foreign key FK_BEBIDAS_PERTENECE_INVENTARIO;
```

```
alter table CLIENTE
```

```
drop foreign key FK_CLIENTE_PERTENECE_RESTAURANTE;
```

```
alter table DETALLE_FACTURA
```

```
drop foreign key FK_DETALLE_FACTURA_TIENE_FACTURA;
```

```
alter table DETALLE_FACTURA
```

```
drop foreign key FK_DETALLE_FACTURA_TIENE_PEDIDO;
```

```
alter table DETALLE_PEDIDO
```

```
drop foreign key FK_DETALLE_PEDIDO_TIENE_PEDIDO;
```

```
alter table DETALLE_PEDIDO
```

```
drop foreign key FK_DETALLE_PLATO_TIENE_PLATO;
```

```
alter table EMPLEADO
```

```
drop foreign key FK_EMPLEADO_TIENE_CARGO;
```

```
alter table EMPLEADO
```

```
drop foreign key FK_EMPLEADO_PERTENECE_RESTAURANTE;
```

```
alter table FACTURA
```

```
drop foreign key FK_FACTURA_TIENE_CLIENTE;
```

```
alter table INGREDIENTE
```

```
drop foreign key FK_INGREDIENTE_PERTENECE_INVENTARIO;
```

```
alter table INVENTARIO
```

```
drop foreign key FK_INVENTARIO_PERTENECE_RESTAURANTE;
```

```
alter table MATERIAL_LIMPIEZA
```

```
drop foreign key FK_MATERIAL_LIMPIEZA_PERTENECE_INVENTARIO;
```

```
alter table PEDIDO
```

```
drop foreign key FK_PEDIDO_TIENE_CLIENTE;
```

```
alter table PLATO
```

```
drop foreign key FK_PLATO_PERTENECE_RESTAURANTE;
```

```
alter table RECETA
```

```
drop foreign key FK_RECETA_TIENE_INGREDIENTE;
```

```
alter table RECETA
```

```
drop foreign key FK_RECETA_PLATO_PERTENECE_PLATO;
```

```
alter table RECETA
```

```
drop foreign key FK_RECETA_TIENE_PREPARACION;
```

```
alter table RESERVA
```

```
drop foreign key FK_RESERVA_TIENE_CLIENTE;
```

```
alter table BEBIDAS
```

```
drop foreign key FK_BEBIDAS_PERTENECE_INVENTARIO;
```

```
drop table if exists BEBIDAS;
```

```
drop table if exists CARGO;
```

```
alter table CLIENTE
```

```
drop foreign key FK_CLIENTE_PERTENECE_RESTAURANTE;
```

```
drop table if exists CLIENTE;
```

```
alter table DETALLE_FACTURA
```

```
drop foreign key FK_DETALLE_FACTURA_TIENE_FACTURA;
```

```
alter table DETALLE_FACTURA
```

```
drop foreign key FK_DETALLE_FACTURA_TIENE_PEDIDO;
```

```
drop table if exists DETALLE_FACTURA;
```

```
alter table DETALLE_PEDIDO
```

```
drop foreign key FK_DETALLE_PLATO_TIENE_PLATO;
```

```
alter table DETALLE_PEDIDO
```

```
drop foreign key FK_DETALLE_PEDIDO_TIENE_PEDIDO;
```

```
drop table if exists DETALLE_PEDIDO;
```

```
alter table EMPLEADO
```

```
drop foreign key FK_EMPLEADO_PERTENECE_RESTAURANTE;
```

alter table EMPLEADO

drop foreign key FK_EMPLEADO_TIENE_CARGO;

drop table if exists EMPLEADO;

alter table FACTURA

drop foreign key FK_FACTURA_TIENE_CLIENTE;

drop table if exists FACTURA;

alter table INGREDIENTE

drop foreign key FK_INGREDIENTE PERTENECE_INVENTARIO;

drop table if exists INGREDIENTE;

alter table INVENTARIO

drop foreign key FK_INVENTARIO_PERTENECE_RESTAURANTE;

drop table if exists INVENTARIO;

alter table MATERIAL_LIMPIEZA

drop foreign key FK_MATERIAL_LIMPIEZA_PERTENECE_INVENTARIO;

drop table if exists MATERIAL_LIMPIEZA;

alter table PEDIDO

drop foreign key FK_PEDIDO_TIENE_CLIENTE;

drop table if exists PEDIDO;

alter table PLATO

```
drop foreign key FK_PLATO_PERTENECE_RESTAURANTE;
```

```
drop table if exists PLATO;
```

```
drop table if exists PREPARACION;
```

```
alter table RECETA
```

```
drop foreign key FK_RECETA_TIENE_PREPARACION;
```

```
alter table RECETA
```

```
drop foreign key FK_RECETA_PLATO_PERTENECE_PLATO;
```

```
alter table RECETA
```

```
drop foreign key FK_RECETA_TIENE_INGREDIENTE;
```

```
drop table if exists RECETA;
```

```
alter table RESERVA
```

```
drop foreign key FK_RESERVA_TIENE_CLIENTE;
```

```
drop table if exists RESERVA;
```

```
drop table if exists RESTAURANTE;
```

```
/*=====*/
```

```
/* Table: BEBIDAS */
```

```
/*=====*/
```

```
create table BEBIDAS
```

```
(
```

```
    BDS_CODIGO      int not null comment "
```

```
    INV_CODIGO      char(20) comment 'Atributo que contiene los numeros de  
cedula o ruc que definen la unicidad de la tupla',
```

```
    BDS_NOMBRE      char(20) comment "
```

```
    BDS_DESCRIPCION text comment "
```

```
    BDS_CANTIDAD    int comment "
```

```
    primary key (BDS_CODIGO)
```

```
);
```

```
alter table BEBIDAS comment 'Contiene informaci n de los equipos_utencillos del
restauran';
```

```
/*=====*/
```

```
/* Table: CARGO */
```

```
/*=====*/
```

```
create table CARGO
```

```
(
```

```
  CAR_CODIGO      char(20) not null comment "",
```

```
  CAR_DESCRIPCION text comment "",
```

```
  primary key (CAR_CODIGO)
```

```
);
```

```
alter table CARGO comment 'Contiene informaci n del cargo del los empleados del
restaur';
```

```
/*=====*/
```

```
/* Table: CLIENTE */
```

```

/*=====*/

create table CLIENTE

(

    CLI_CEDULA      char(13) not null comment 'Atributo que contiene los numeros
de cedula o ruc que definen la unicidad de la tupla',

    RES_CODIGO     char(20) not null comment 'Atributo que contiene los codigo
del proyecto que definen la unicidad de la tupla',

    CLI_NOMBRE     char(20) comment "",

    CLI_APELLIDO   char(20) comment "",

    CLI_DIRRECCION char(40) comment "",

    CLI_TELEFONO   char(10) comment "",

    CLI_CORREO     char(20) comment "",

    primary key (CLI_CEDULA)

);

alter table CLIENTE comment 'Contiene informacion del cliente en el restaurante';

```

```
/*=====*/
```

```
/* Table: DETALLE_FACTURA */
```

```
/*=====*/
```

```

create table DETALLE_FACTURA

(

    FAC_NUMERO      char(15) comment 'Atributo que contiene los numeros de
cedula o ruc que definen la unicidad de la tupla',

    PDD_CODIGO      int comment 'Atributo que contiene los numeros de cedula o
ruc que definen la unicidad de la tupla'

);

/*=====*/

/* Table: DETALLE_PEDIDO */

/*=====*/

create table DETALLE_PEDIDO

(

    PLT_CODIGO      varchar(13) comment 'Atributo que contiene los numeros de
cedula o ruc que definen la unicidad de la tupla',

    PDD_CODIGO      int comment 'Atributo que contiene los numeros de cedula o
ruc que definen la unicidad de la tupla'

);

/*=====*/

```

```

/* Table: EMPLEADO                                     */

/*=====*/

create table EMPLEADO

(

    CAR_CODIGO      char(20) not null comment "",

    EMP_CODIGO      char(20) not null comment 'Atributo que contiene la
informacion del codigo del empleado',

    RES_CODIGO      char(20) not null comment 'Atributo que contiene los codigo
del proyecto que definen la unicidad de la tupla',

    EMP_CEDULA      char(10) comment 'Atributo que contiene la informacion de la
cedula del empleado',

    EMP_NOMBRE      char(20) comment 'Atributo que contiene la informacion del
nombre del empleado',

    EMP_APELLIDO    char(20) comment "",

    EMP_DIRECCION   char(40) comment 'Atributo que contiene la informacion de
la direccion del empleado',

    EMP_TELEFONO    char(20) comment "",

    EMP_CORREO      char(20) comment "",

    primary key (CAR_CODIGO, EMP_CODIGO)

);

```

```
alter table EMPLEADO comment 'Contiene informaci n de los empleados en el
restaurante';
```

```
/*=====*/
```

```
/* Table: FACTURA */
```

```
/*=====*/
```

```
create table FACTURA
(
    FAC_NUMERO      char(15) not null comment 'Atributo que contiene los
numeros de cedula o ruc que definen la unicidad de la tupla',
    CLI_CEDULA      char(13) not null comment 'Atributo que contiene los numeros
de cedula o ruc que definen la unicidad de la tupla',
    FAC_FECHA       datetime comment '',
    primary key (FAC_NUMERO)
);
```

```
alter table FACTURA comment 'Contiene informacion de factura de cada pedido';
```

```
/*=====*/
```

```

/* Table: INGREDIENTE                                     */

/*=====*/

create table INGREDIENTE

(

    ING_CODIGO      int not null comment 'Atributo que contiene los numeros de
cedula o ruc que definen la unicidad de la tupla',

    INV_CODIGO      char(20) not null comment 'Atributo que contiene los numeros
de cedula o ruc que definen la unicidad de la tupla',

    ING_NOMBRE      char(20) comment 'Contiene el nombre del producto',

    ING_DESCRIPCION text comment 'Contiene la descripci◊n del producto',

    ING_CANTIDAD    int comment '',

    primary key (ING_CODIGO)

);

alter table INGREDIENTE comment 'Contiene informacion del ingrediente de cada
inventario';

```

```

/*=====*/

/* Table: INVENTARIO                                     */

/*=====*/

```

```

create table INVENTARIO
(
    INV_CODIGO      char(20) not null comment 'Atributo que contiene los numeros
de cedula o ruc que definen la unicidad de la tupla',
    RES_CODIGO      char(20) not null comment 'Atributo que contiene los codigo
del proyecto que definen la unicidad de la tupla',
    INV_CANTIDAD    int comment '',
    INV_ESTADO      int comment '',
    primary key (INV_CODIGO)
);

```

```

alter table INVENTARIO comment 'Contiene informacion del inventario de cada
restaurante';

```

```

/*=====*/

```

```

/* Table: MATERIAL_LIMPIEZA */

```

```

/*=====*/

```

```

create table MATERIAL_LIMPIEZA

```

```

(
    MLZ_CODIGO      int not null comment '',

```

```
    INV_CODIGO      char(20) comment 'Atributo que contiene los numeros de
cedula o ruc que definen la unicidad de la tupla',
```

```
    MLZ_NOMBRE      char(20) comment '',
```

```
    MLZ_DESCIPCION  text comment '',
```

```
    MLZ_CANTIDAD    int comment '',
```

```
    primary key (MLZ_CODIGO)
```

```
);
```

```
alter table MATERIAL_LIMPIEZA comment 'Contiene informaci n de los
material_limpieza del restaurant';
```

```
/*=====*/
```

```
/* Table: PEDIDO */
```

```
/*=====*/
```

```
create table PEDIDO
```

```
(
```

```
    PDD_CODIGO      int not null comment 'Atributo que contiene los numeros de
cedula o ruc que definen la unicidad de la tupla',
```

```
    CLI_CEDULA      char(13) not null comment 'Atributo que contiene los numeros
de cedula o ruc que definen la unicidad de la tupla',
```

```
        PDD_FECHA      datetime comment 'Atributo en el que se guarda el numero de
telefono de cada cliente',
```

```
        PDD_TOTAL      float comment ",
```

```
        primary key (PDD_CODIGO)
```

```
);
```

```
alter table PEDIDO comment 'Contiene informacion del pedido de cada cliente';
```

```
/*=====*/
```

```
/* Table: PLATO */
```

```
/*=====*/
```

```
create table PLATO
```

```
(
```

```
        PLT_CODIGO      varchar(13) not null comment 'Atributo que contiene los
numeros de cedula o ruc que definen la unicidad de la tupla',
```

```
        RES_CODIGO      char(20) not null comment 'Atributo que contiene los codigo
del proyecto que definen la unicidad de la tupla',
```

```
        PLT_DESCRIPCION  text comment ",
```

```
        PLT_PRECIO      float comment ",
```

```
        primary key (PLT_CODIGO)
```

```
);
```

```
alter table PLATO comment 'Contiene informacion de los platos del restaurante';
```

```
/*=====*/
```

```
/* Table: PREPARACION */
```

```
/*=====*/
```

```
create table PREPARACION
```

```
(
```

```
PRP_ID char(6) not null comment "",
```

```
PRP_INSTRUCCIONES text comment "",
```

```
primary key (PRP_ID)
```

```
);
```

```
alter table PREPARACION comment 'Contiene prformaci♦n de los preparaci♦ns del  
restaurante
```

```
-&';
```

```
/*=====*/
```

```

/* Table: RECETA                                     */

/*=====*/

create table RECETA

(

    PRP_ID          char(6) comment "",

    PLT_CODIGO      varchar(13) comment 'Atributo que contiene los numeros de
cedula o ruc que definen la unicidad de la tupla',

    ING_CODIGO      int comment 'Atributo que contiene los numeros de cedula o
ruc que definen la unicidad de la tupla',

    RCT_DESCRIPCION text comment ""

);

/*=====*/

/* Table: RESERVA                                     */

/*=====*/

create table RESERVA

(

    RSV_CODIGO      int not null comment "",

    CLI_CEDULA      char(13) not null comment 'Atributo que contiene los numeros
de cedula o ruc que definen la unicidad de la tupla',

```

```
RSV_FECHA      datetime comment 'Atributo que contiene los numeros de
cedula o ruc que definen la unicidad de la tupla',
```

```
RSV_CANT_PERS  int comment ",
```

```
primary key (RSV_CODIGO)
```

```
);
```

```
alter table RESERVA comment 'Contiene informacion de la reserva de cada cliente ';
```

```
/*=====*/
```

```
/* Table: RESTAURANTE */
```

```
/*=====*/
```

```
create table RESTAURANTE
```

```
(
```

```
RES_CODIGO     char(20) not null comment 'Atributo que contiene los codigo
del proyecto que definen la unicidad de la tupla',
```

```
RES_NOMBRE     char(20) comment 'Atributo que contiene el nombre del
proyecto',
```

```
RES_DESCRIPCION char(40) comment 'Atributo que contiene la descripcion
del proyecto',
```

```
RES_TELEFONO char(10) comment 'Atributo que contiene la informacion del
presupuesto del proyecto',
```

```
RES_DIRECCION char(40) comment 'Atributo que contiene la informacion del
numero de horas del proyecto',
```

```
primary key (RES_CODIGO)
```

```
);
```

```
alter table RESTAURANTE comment 'Entidad en la que se registra informacion del
restaurante';
```

```
alter table BEBIDAS add constraint FK_BEBIDAS_PERTENECE_INVENTARIO
foreign key (INV_CODIGO)
```

```
references INVENTARIO (INV_CODIGO) on delete restrict on update restrict;
```

```
alter table CLIENTE add constraint FK_CLIENTE_PERTENECE_RESTAURANTE
foreign key (RES_CODIGO)
```

```
references RESTAURANTE (RES_CODIGO) on delete restrict on update restrict;
```

```
alter table DETALLE_FACTURA add constraint
FK_DETALLE_FACTURA_TIENE_FACTURA foreign key (FAC_NUMERO)
```

```
references FACTURA (FAC_NUMERO) on delete restrict on update restrict;
```

```
alter table DETALLE_FACTURA add constraint  
FK_DETALLE_FACTURA_TIENE_PEDIDO foreign key (PDD_CODIGO)  
  
references PEDIDO (PDD_CODIGO) on delete restrict on update restrict;
```

```
alter table DETALLE_PEDIDO add constraint  
FK_DETALLE_PEDIDO_TIENE_PEDIDO foreign key (PDD_CODIGO)  
  
references PEDIDO (PDD_CODIGO) on delete restrict on update restrict;
```

```
alter table DETALLE_PEDIDO add constraint FK_DETALLE_PLATO_TIENE_PLATO  
foreign key (PLT_CODIGO)  
  
references PLATO (PLT_CODIGO) on delete restrict on update restrict;
```

```
alter table EMPLEADO add constraint FK_EMPLEADO_TIENE_CARGO foreign key  
(CAR_CODIGO)  
  
references CARGO (CAR_CODIGO) on delete restrict on update restrict;
```

```
alter table EMPLEADO add constraint  
FK_EMPLEADO_PERTENECE_RESTAURANTE foreign key (RES_CODIGO)  
  
references RESTAURANTE (RES_CODIGO) on delete restrict on update restrict;
```

alter table FACTURA add constraint FK_FACTURA_TIENE_CLIENTE foreign key
(CLI_CEDULA)

references CLIENTE (CLI_CEDULA) on delete restrict on update restrict;

alter table INGREDIENTE add constraint
FK_INGREDIENTE_PERTENECE_INVENTARIO foreign key (INV_CODIGO)

references INVENTARIO (INV_CODIGO) on delete restrict on update restrict;

alter table INVENTARIO add constraint
FK_INVENTARIO_PERTENECE_RESTAURANTE foreign key (RES_CODIGO)

references RESTAURANTE (RES_CODIGO) on delete restrict on update restrict;

alter table MATERIAL_LIMPIEZA add constraint
FK_MATERIAL_LIMPIEZA_PERTENECE_INVENTARIO foreign key (INV_CODIGO)

references INVENTARIO (INV_CODIGO) on delete restrict on update restrict;

alter table PEDIDO add constraint FK_PEDIDO_TIENE_CLIENTE foreign key
(CLI_CEDULA)

references CLIENTE (CLI_CEDULA) on delete restrict on update restrict;

alter table PLATO add constraint FK_PLATO_PERTENECE_RESTAURANTE foreign
key (RES_CODIGO)

references RESTAURANTE (RES_CODIGO) on delete restrict on update restrict;

alter table RECETA add constraint FK_RECETA_TIENE_INGREDIENTE foreign key
(ING_CODIGO)

references INGREDIENTE (ING_CODIGO) on delete restrict on update restrict;

alter table RECETA add constraint FK_RECETA_PLATO_PERTENECE_PLATO
foreign key (PLT_CODIGO)

references PLATO (PLT_CODIGO) on delete restrict on update restrict;

alter table RECETA add constraint FK_RECETA_TIENE_PREPARACION foreign key
(PRP_ID)

references PREPARACION (PRP_ID) on delete restrict on update restrict;

alter table RESERVA add constraint FK_RESERVA_TIENE_CLIENTE foreign key
(CLI_CEDULA)

references CLIENTE (CLI_CEDULA) on delete restrict on update restrict;

Anexo B: Capturas de pantalla

Figura 71

Interfaz dashboard gestión de restaurante



Autor: Cadena, S. (2023)

Figura 72

Interfaz dashboard gestión de pedidos - visualización

The screenshot displays a web application interface for order management. The header includes the brand name 'LOGOPIZZA' and a user profile icon. The left sidebar lists navigation items: 'INICIO', 'Dashboard', and 'INTERFACE' with sub-items 'Empleados', 'Clientes', 'Inventario', and 'Pedidos'. The main content area is titled 'Dashboard' and 'Gestión de Pedidos'. It features a table with the following data:

| CODIGO PEDIDO | CEDULA CLIENTE | FECHA | TOTAL | Botones |
|---------------|----------------|---------------------|-------|---------------------|
| 1 | 1111111111 | 2023-05-30 21:03:27 | 15 | Eliminar Actualizar |
| 2 | 2222222222 | 2023-05-30 21:03:44 | 15 | Eliminar Actualizar |

At the bottom of the page, there is a footer with the text 'Logged in as: AENSXX', 'Copyright © LOGOPIZZA 2023', and a link for 'Políticas de Privacidad · Terminos y Condiciones'.

Autor: Cadena, S. (2023)

Figura 73

Interfaz dashboard gestión de pedidos - ingreso pedido

The image shows a web application interface for order management. On the left is a dark sidebar with the logo 'LOGOPIZZA' and a menu. The main content area is titled 'Dashboard' and 'Gestión de Pedidos'. It features a form titled 'Agregar nuevo Pedido' with the following fields: 'PEDIDO CODIGO: PEDIDO CODIGO', 'Pedido Fecha: Pedido Fecha', 'CEDULA CLIENTE: CEDULA CLIENTE', and 'Nombre: Nombre empleado'. An 'Agregar' button is at the bottom of the form. The footer contains 'Logged in as: AENSXX', 'Copyright © LOGOPIZZA 2023', and a link to 'Políticas de Privacidad · Terminos y & Condiciones'.

LOGOPIZZA

INICIO

- Dashboard

INTERFACE

- Empleados
- Cientes
- Inventario
- Pedidos
- Platos
- Reservas

Dashboard

Gestión de Pedidos

Agregar nuevo Pedido

PEDIDO CODIGO:

Pedido Fecha:

CEDULA CLIENTE:

Nombre:

Logged in as: AENSXX

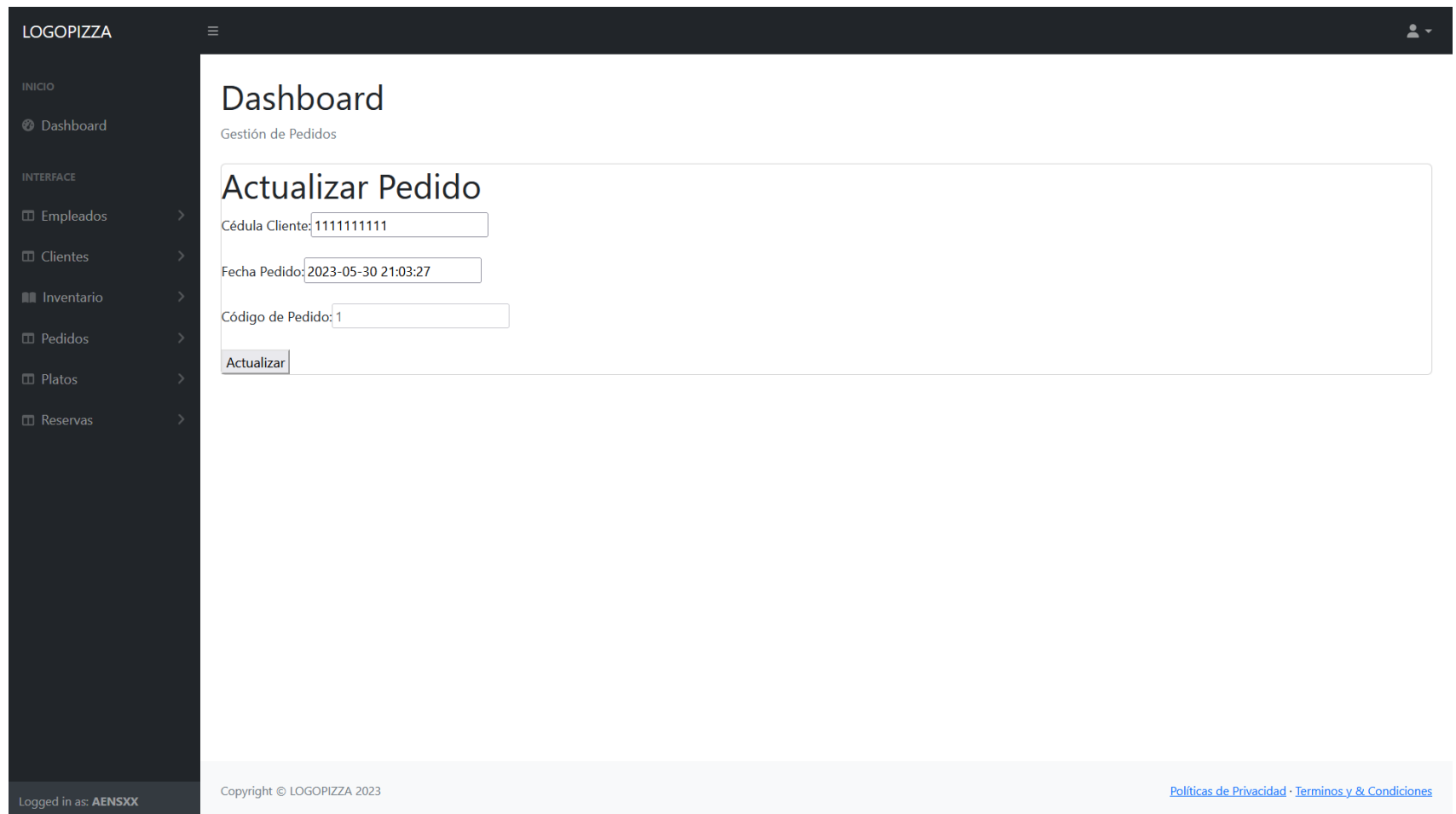
Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y & Condiciones](#)

Autor: Cadena, S. (2023)

Figura 74

Interfaz dashboard gestión de pedidos - Actualización de pedido



Autor: Cadena, S. (2023)

Figura 75

Interfaz dashboard gestión de platos - Ver plato

The screenshot displays a web dashboard for 'LOGOPIZZA'. The main content area is titled 'Dashboard' and 'Gestión de Platos'. It features a table with the following data:

| PLATO CODIGO | RESTAURANTE CODIGO | DESCRIPCION | PRECIO | Botones |
|--------------|--------------------|------------------------------|--------|---------------------|
| COMBO1 | 001 | PIZZA DE JAMON CON COCA-COLA | 12 | Eliminar Actualizar |
| COMBO2 | 001 | PIZZA DE JAMON CON SPRITE | 12 | Eliminar Actualizar |
| COMBO3 | 001 | PIZZA DE JAMON CON FANTA | 12 | Eliminar Actualizar |

The dashboard also includes a sidebar with navigation links: INICIO, Dashboard, INTERFACE, Empleados, Clientes, Inventario, and Pedidos. At the bottom, it shows 'Logged in as: AENSXX', 'Copyright © LOGOPIZZA 2023', and links for 'Políticas de Privacidad' and 'Terminos y Condiciones'.

Autor: Cadena, S. (2023)

Figura 76

Interfaz dashboard gestión de platos - Ingresar plato

The image shows a web application interface for managing dishes. On the left is a dark sidebar with the logo 'LOGOPIZZA' and a menu. The main content area is titled 'Dashboard' and 'Gestión de Platos'. It features a form titled 'Agregar nuevo Plato' with three input fields: 'CODIGO PLATO' (containing 'Codigo Plato'), 'PLATO DESCRIPCION' (containing 'Plato Descripción'), and 'PLATO PRECIO' (containing 'Plato Precio'). Below these fields is an 'Agregar' button. At the bottom of the page, there is a footer with 'Logged in as: AENSXX', 'Copyright © LOGOPIZZA 2023', and a link for 'Políticas de Privacidad · Terminos y & Condiciones'.

Autor: Cadena, S. (2023)

Figura 77

Interfaz dashboard gestión de platos - Actualizar plato

The screenshot displays the LOGOPIZZA dashboard interface. On the left is a dark sidebar with navigation options: INICIO (with 'Dashboard' selected), INTERFACE (with 'Empleados', 'Clientes', 'Inventario', 'Pedidos', 'Platos', and 'Reservas'), and a user profile icon at the top right. The main content area is titled 'Dashboard' and 'Gestión de Platos'. A form titled 'Actualizar Plato' contains the following fields: 'RestauranteCodigo' (001), 'DescripciónPlato' (PIZZA DE JAMON CON COC), 'Precio' (12), and 'CodigoPlato' (COMBO1). An 'Actualizar' button is located at the bottom of the form. The footer includes 'Logged in as: AENSXX', 'Copyright © LOGOPIZZA 2023', and a link to 'Políticas de Privacidad · Terminos y & Condiciones'.

Autor: Cadena, S. (2023)

Figura 78

Interfaz dashboard gestión de clientes - Ver clientes

Dashboard

Gestión de Clientes

| CEDULA | CODIGO | Nombre | Apellido | Direccion | Telefono | Correo Electronico | Botones |
|------------|--------|-------------|----------|---------------|------------|--------------------|---------------------|
| 1 | 001 | qweasdasdas | qwesw | av la prensa | 999 | qwe@qwe.com | Eliminar Actualizar |
| 1111111111 | 001 | NORA | MOLINA | AV LA PRENSA | 0000000000 | NORA@NORA.COM | Eliminar Actualizar |
| 1709194771 | 001 | CECILIA | MOLINA | AV. LA PRENSA | 123 | NORITA@NORITA.COM | Eliminar Actualizar |
| 2222222222 | 001 | CARMEN | MOLINA | AV LA PRENSA | 0000000000 | CARMEN@CARMEN.COM | Eliminar Actualizar |
| 9999999999 | 001 | NORITA | MOLINA | AV. LA PRENSA | 123 | NORITA@NORITA.COM | Eliminar Actualizar |

Logged in as: AENSXX

Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y Condiciones](#)

Autor: Cadena, S. (2023)

Figura 79

Interfaz dashboard gestión de clientes - Ingresar cliente

The image shows a web application interface for a pizza shop named LOGOPIZZA. The interface is divided into a dark sidebar on the left and a main content area on the right. The sidebar contains a navigation menu with the following items: INICIO, Dashboard, INTERFACE, Empleados, Clientes, Inventario, Pedidos, Platos, and Reservas. The main content area displays the title 'Dashboard' and 'Gestión de Clientes'. Below this, there is a form titled 'Agregar nuevo Cliente' with the following fields: Cedula (Cedula Cliente), Nombre (Nombre Cliente), Apellido (Apellido Cliente), Direccion (Dirección), Telefono (Telefono Cliente), and Correo (Correo Cliente). An 'Agregar' button is located at the bottom of the form. At the bottom of the page, there is a footer with the text 'Logged in as: AENSXX', 'Copyright © LOGOPIZZA 2023', and a link to 'Políticas de Privacidad · Terminos y & Condiciones'.

LOGOPIZZA

INICIO

Dashboard

INTERFACE

Empleados

Clientes

Inventario

Pedidos

Platos

Reservas

Dashboard

Gestión de Clientes

Agregar nuevo Cliente

Cedula:

Nombre:

Apellido:

Direccion:

Telefono:

Correo:

Logged in as: AENSXX

Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y & Condiciones](#)

Autor: Cadena, S. (2023)

Figura 80

Interfaz dashboard gestión de clientes - Actualizar cliente

The screenshot shows a web application interface for updating a customer. On the left is a dark sidebar with the logo 'LOGOPIZZA' and a menu with categories: 'INICIO' (containing 'Dashboard'), 'INTERFACE' (containing 'Empleados', 'Clientes', 'Inventario', 'Pedidos', 'Platos', 'Reservas'), and a user profile icon. The main content area is titled 'Dashboard' with a subtitle 'Gestión de Clientes'. Below this is a form titled 'Actualizar Cliente' with the following fields: 'NOMBRE: CARMEN', 'Apellido: MOLINA', 'Dirección Cliente: AV LA PRENSA', 'Número de Teléfono: 0000000000', 'Correo Electronico: CARMEN@CARMEN.COM', and 'CEDULA: 2222222222'. An 'Actualizar' button is located at the bottom of the form. The footer contains 'Logged in as: AENSXX', 'Copyright © LOGOPIZZA 2023', and a link for 'Políticas de Privacidad · Terminos y & Condiciones'.

Autor: Cadena, S. (2023)

Figura 81

Interfaz dashboard gestión de empleados - Ver empleados

LOGOPIZZA

Dashboard

Gestión de Empleado

| CEDULA | CODIGO | CODIGO EMP | Nombre | Apellido | Direccion | Telefono | Correo Electronico | Botones |
|------------|--------|------------|--------|-----------|----------------|----------|--------------------|---------------------|
| 1722941455 | 001 | 1 | AENSXX | CADEN | AV. LA PRENSA | 111 | AENS@AENS.COM | Eliminar Actualizar |
| 1701253070 | 001 | 2 | BSK | CADE | AV LA PRENSA | 2 | BSK@BSK.COM | Eliminar Actualizar |
| 9999999999 | 001 | 4 | NORITA | MOLINA | AV. LA PRENSA | 123 | NORITA@NORITA.COM | Eliminar Actualizar |
| 1722941505 | 001 | 3 | ELIO | CADE | AV LA PRENSA | 3 | ELIO@ELIO.COM | Eliminar Actualizar |
| 444 | 001 | JoMar3 | Jose | Martinezx | AV. LA PRENSAs | 123 | ssjo@jo.com | Eliminar Actualizar |

Logged in as: AENSXX

Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y Condiciones](#)

Autor: Cadena, S. (2023)

Figura 82

Interfaz dashboard gestión de empleados - Agregar empleado

The image shows a web application interface for employee management. On the left is a dark sidebar with the logo 'LOGOPIZZA' and a menu with categories 'INICIO' and 'INTERFACE'. Under 'INTERFACE', there are links for 'Dashboard', 'Empleados', 'Clientes', 'Inventario', 'Pedidos', 'Platos', and 'Reservas'. The main content area is titled 'Dashboard' and 'Gestión de Empleado'. It features a form titled 'Agregar nuevo Empleado' with the following fields: 'cargo' (a dropdown menu set to 'ENCARGADO INVENTARIO'), 'Cedula' (text input 'Cedula empleado'), 'empleado codigo' (text input 'empleado codigo'), 'Nombre' (text input 'Nombre empleado'), 'Apellido' (text input 'Apellido empleado'), 'Direccion' (text input 'Dirección'), 'Telefono' (text input 'Telefono empleado'), and 'Correo' (text input 'Correo empleado'). A grey 'Agregar' button is at the bottom of the form. The footer contains 'Logged in as: AENSXX', 'Copyright © LOGOPIZZA 2023', and a link for 'Políticas de Privacidad · Terminos y & Condiciones'.

Autor: Cadena, S. (2023)

Figura 83

Interfaz dashboard gestión de empleados - Actualizar empleado

LOGOPIZZA

Dashboard

Gestión de Empleado

Actualizar Empleado

NOMBRE:

Apellido:

Dirección Empleado:

Número de Teléfono:

Correo Electronico:

CEDULA:

Logged in as: AENSXX

Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y Condiciones](#)

Autor: Cadena, S. (2023)

Figura 84

Interfaz dashboard gestión de reservas - Ver reservas

The screenshot displays a web application interface for reservation management. On the left is a dark sidebar with the logo 'LOGOPIZZA' and a menu containing 'INICIO', 'Dashboard', and 'INTERFACE' with sub-items: 'Empleados', 'Clientes', 'Inventario', and 'Pedidos'. The main content area is titled 'Dashboard' and 'Gestión de Reservas'. It features a table with the following data:

| CODIGO RESERVA | CEDULA CLIENTE | FECHA RESERVA | CANTIDAD PERSONAS | Botones | |
|----------------|----------------|---------------------|-------------------|----------|------------|
| 1 | 1111111111 | 2023-05-30 21:06:14 | 5 | Eliminar | Actualizar |
| 2 | 2222222222 | 2023-05-30 21:06:24 | 6 | Eliminar | Actualizar |

At the bottom of the page, there is a footer with 'Logged in as: AENSXX', 'Copyright © LOGOPIZZA 2023', and a link for 'Políticas de Privacidad · Terminos y & Condiciones'.

Autor: Cadena, S. (2023)

Figura 85

Interfaz dashboard gestión de reserva - Agregar reserva

LOGOPIZZA

INICIO

- Dashboard

INTERFACE

- Empleados
- Clientes
- Inventario
- Pedidos
- Platos
- Reservas

Dashboard

Gestión de Reservas

Agregar nueva Reserva

Codigo Reserva:

Cedula Cliente:

Fecha de Reserva:

Cantidad Personas:

Logged in as: AENSXX

Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y Condiciones](#)

Autor: Cadena, S. (2023)

Figura 86

Interfaz dashboard gestión de reserva - Actualizar reserva

LOGOPIZZA

INICIO

- Dashboard

INTERFACE

- Empleados
- Clientes
- Inventario
- Pedidos
- Platos
- Reservas

Dashboard

Gestión de Empleado

Actualizar Reserva

CODIGO RESERVA: 1

CEDULA CLIENTE: 1111111111

FECHA DE RESERVA: 2023-05-30 21:06:14

CANTIDAD DE PERSONAS: 5

Actualizar

Logged in as: AENSXX

Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y Condiciones](#)

Autor: Cadena, S. (2023)

Figura 87

Interfaz dashboard gestión de materiales de limpieza - Ver materiales de limpieza

LOGOPIZZA

INICIO

- Dashboard

INTERFACE

- Empleados
- Cientes
- Inventario
- Pedidos

Dashboard

Gestión de Inventario - Materiales de Limpieza

| CODIGO MATERIAL | CODIGO INVENTARIO | NOMBRE MATERIAL | DESCIPCION | CANTIDAD | Botones |
|-----------------|-------------------|-----------------|---------------------------------|----------|---------------------|
| 1 | MATERIALES | CLORO | LITRO DE CLORO | 10 | Eliminar Actualizar |
| 2 | MATERIALES | DETERGENTE | GALON DE DETERGENTE | 12 | Eliminar Actualizar |
| 3 | MATERIALES | DESINFECTANTE | GALON DESINFECTANTE | 10 | Eliminar Actualizar |
| 4 | MATERIALES | FOCO | Foco para luces del restaurante | 12 | Eliminar Actualizar |

Logged in as: AENSXX

Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y Condiciones](#)

Autor: Cadena, S. (2023)

Figura 88

Interfaz dashboard gestión de materiales de limpieza - Agregar materiales de limpieza

LOGOPIZZA

INICIO

- Dashboard

INTERFACE

- Empleados
- Clientes
- Inventario
- Pedidos
- Platos
- Reservas

Dashboard

Gestión de Inventario - Materiales de Limpieza

Agregar nuevo Material de Limpieza

Código Inventario: **MATERIALES**

Codigo Material:

Nombre Material:

Cantidad Material:

Descripción Material:

Descripción Material:

Logged in as: AENSXX

Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y Condiciones](#)

Autor: Cadena, S. (2023)

Figura 89

Interfaz dashboard gestión de materiales de limpieza - Actualizar materiales de limpieza

LOGOPIZZA

INICIO

- Dashboard

INTERFACE

- Empleados
- Cientes
- Inventario
- Pedidos
- Platos
- Reservas

Dashboard

Inventario de Materiales de Limpieza

Actualiza el Registro de Material de Limpieza

CODIGO MATERIAL:

NOMBRE MATERIAL:

CANTIDAD MATERIAL:

DESCIPCIÓN MATERIAL:

Logged in as: AENSXX

Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y & Condiciones](#)

Autor: Cadena, S. (2023)

Figura 90

Interfaz dashboard gestión de bebidas - Ver bebidas

LOGOPIZZA

Dashboard

Gestión de Inventario - Bebidas

| CODIGO BEBIDA | CODIGO INVENTARIO | NOMBRE BEBIDA | DESCIPCION | CANTIDAD | Botones |
|---------------|-------------------|---------------|------------|----------|---------------------|
| 1 | BEBIDAS | COCA-COLA | 1 LITRO | 10 | Eliminar Actualizar |
| 2 | BEBIDAS | SPRITE | 1 LITRO | 10 | Eliminar Actualizar |
| 3 | BEBIDAS | FANTA | 1 LITRO | 10 | Eliminar Actualizar |

Logged in as: AENSXX

Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y Condiciones](#)

Autor: Cadena, S. (2023)

Figura 91

Interfaz dashboard gestión de bebidas - Agregar bebida

LOGOPIZZA

Dashboard

Gestión de Inventario

Agregar nueva Bebida

Código Inventario: **BEBIDAS**

Codigo Bebida:

Nombre Bebida:

Cantidad Bebida:

Descripción Bebida:

Logged in as: AENSXX

Copyright © LOGOPIZZA 2023

[Políticas de Privacidad · Terminos y & Condiciones](#)

Autor: Cadena, S. (2023)

Figura 92

Interfaz dashboard gestión de bebidas - Actualizar bebida

LOGOPIZZA

INICIO

- Dashboard

INTERFACE

- Empleados
- Clientes
- Inventario
- Pedidos
- Platos
- Reservas

Dashboard

Gestión de Inventario - Bebidas

Actualiza el Registro de Bebida

CODIGO BEBIDA:

NOMBRE BEBIDA:

CANTIDAD BEBIDA:

DESCIPCIÓN BEBIDA:

Logged in as: AENSXX

Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y Condiciones](#)

Autor: Cadena, S. (2023)

Figura 93

Interfaz dashboard gestión de ingrediente - Ver ingrediente

LOGOPIZZA

INICIO

- Dashboard

INTERFACE

- Empleados
- Cientes
- Inventario
- Pedidos

Dashboard

Gestión de Inventario - Ingredientes

| CODIGO INGREDIENTE | CODIGO INVENTARIO | NOMBRE INGREDIENTE | DESCIPCION | CANTIDAD | Botones |
|--------------------|-------------------|--------------------|-------------------------|----------|---------------------|
| 1 | INGREDIENTES | HARINA | HARINA PARA PIZZA EN KG | 10 | Eliminar Actualizar |
| 2 | INGREDIENTES | JAMON | EMBUTIDO DE JAMON | 10 | Eliminar Actualizar |
| 3 | INGREDIENTES | ACEITE | LITRO DE ACEITE | 10 | Eliminar Actualizar |

Logged in as: AENSXX

Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y Condiciones](#)

Autor: Cadena, S. (2023)

Figura 94

Interfaz dashboard gestión de ingrediente - Actualizar ingrediente

The screenshot displays the LOGOPIZZA dashboard interface. On the left is a dark sidebar with navigation options: INICIO (with 'Dashboard' selected), INTERFACE, Empleados, Clientes, Inventario, Pedidos, Platos, and Reservas. The main content area is titled 'Dashboard' and 'Gestión de Inventario - Ingredientes'. The central focus is a form titled 'Actualiza el Registro de Ingrediente' with the following fields: 'CODIGO MATERIAL' (input with value '1'), 'NOMBRE MATERIAL' (input with value 'HARINA'), 'CANTIDAD MATERIAL' (input with value '10'), and 'DESCIPCIÓN MATERIAL' (text area with value 'HARINA PARA PIZZA EN KG'). A grey 'Actualizar' button is positioned below the description field. The footer contains 'Logged in as: AENSXX', 'Copyright © LOGOPIZZA 2023', and a link to 'Políticas de Privacidad · Terminos y & Condiciones'.

Autor: Cadena, S. (2023)

Figura 95

Interfaz dashboard gestión de ingrediente - Agregar ingrediente

LOGOPIZZA

INICIO

- Dashboard

INTERFACE

- Empleados
- Clientes
- Inventario
- Pedidos
- Platos
- Reservas

Dashboard

Gestión de Inventario - Ingredientes

Agregar nuevo Ingrediente

Código Inventario: **INGREDIENTES**

Codigo Ingrediente:

Nombre Ingrediente:

Cantidad Ingrediente:

Descripción Ingrediente:

Descripción Ingrediente:

Logged in as: AENSXX

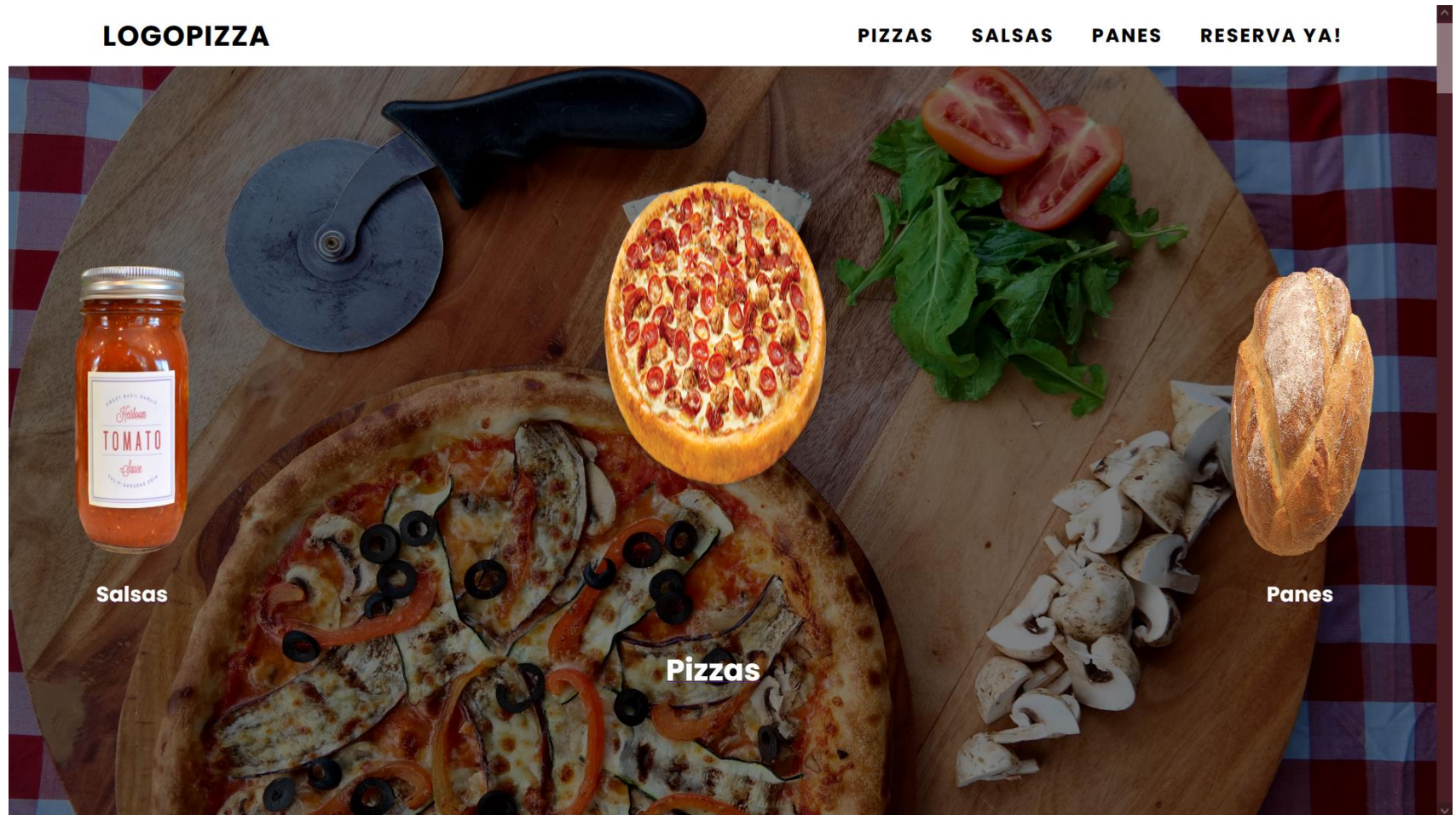
Copyright © LOGOPIZZA 2023

[Políticas de Privacidad](#) · [Terminos y Condiciones](#)

Autor: Cadena, S. (2023)

Figura 96

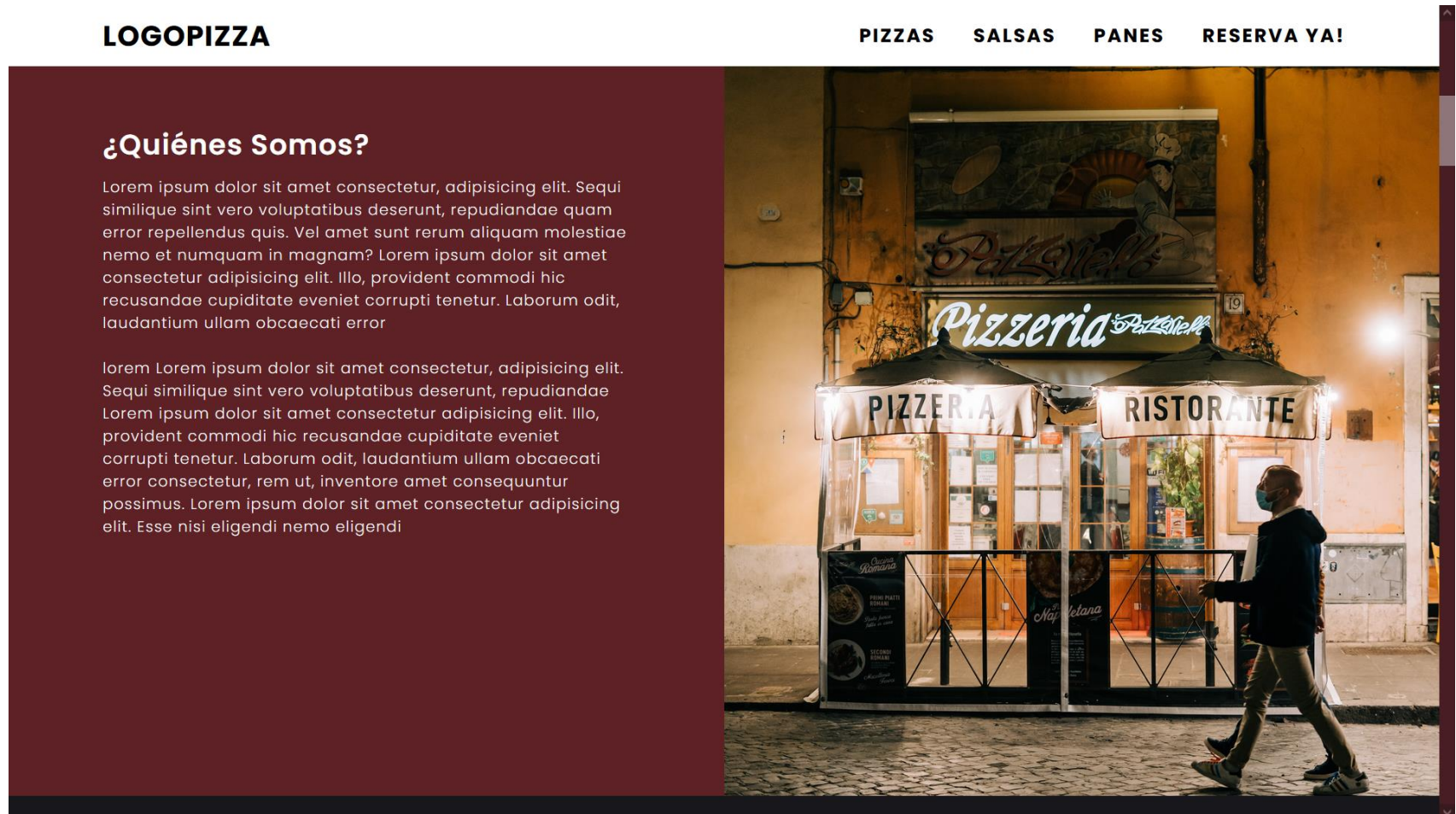
Interfaz página web restaurante – Opciones Platos



Autor: Cadena, S. (2023)

Figura 97

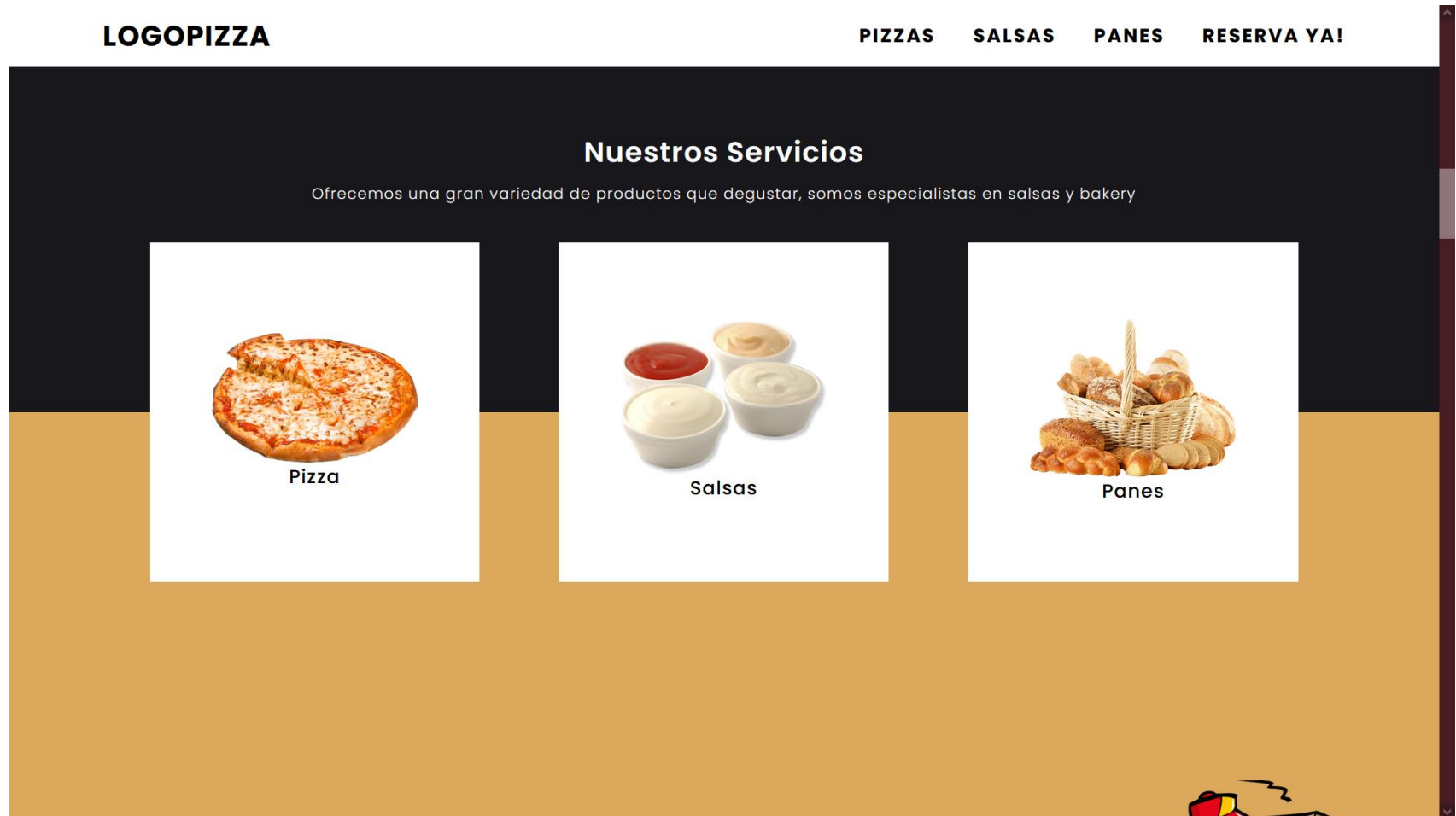
Interfaz página web restaurante - Información restaurante



Autor: Cadena, S. (2023)

Figura 98

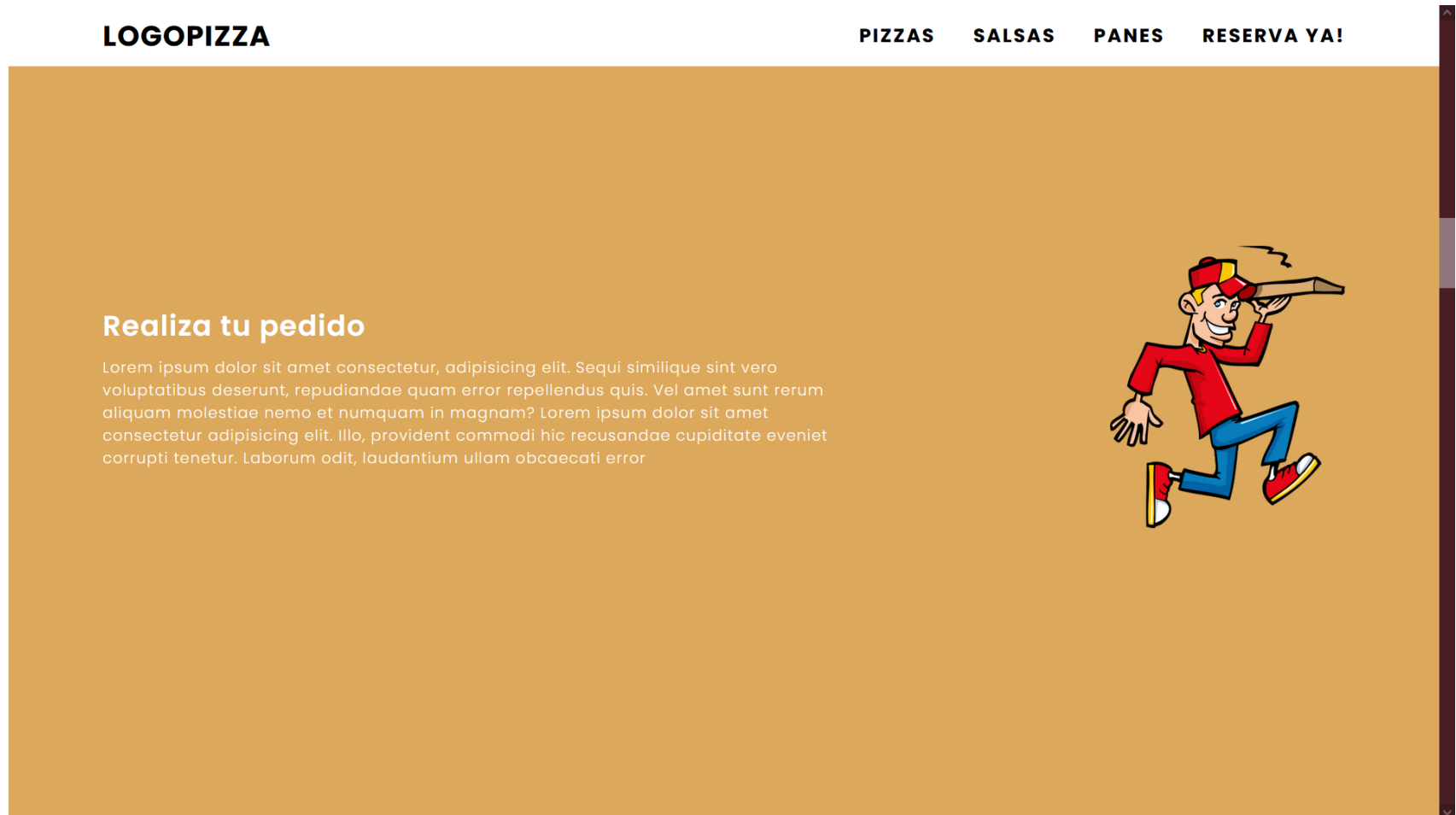
Interfaz página web restaurante - Servicios



Autor: Cadena, S. (2023)

Figura 99

Interfaz página web restaurante - Pedidos



Autor: Cadena, S. (2023)

Figura 100

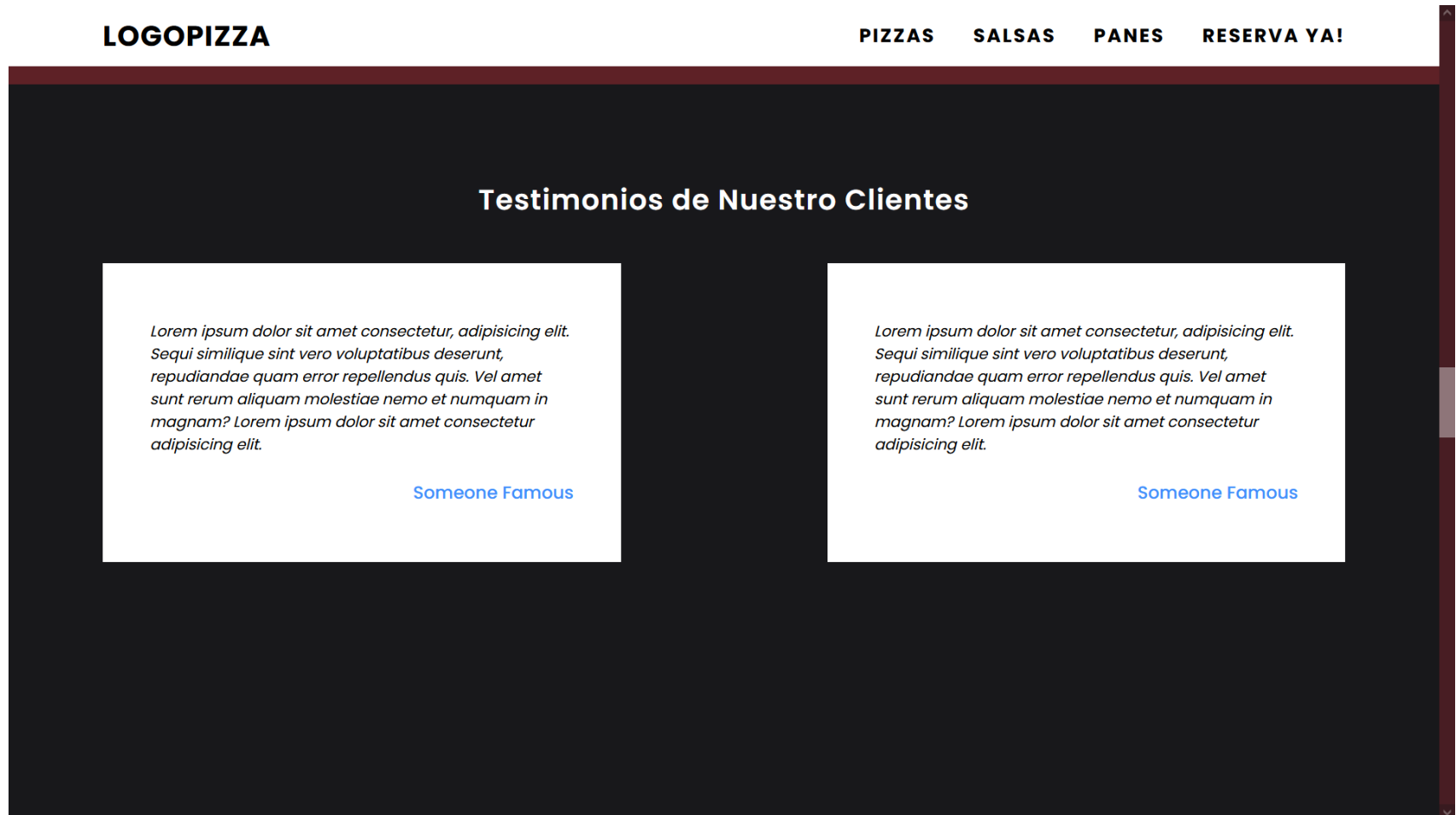
Interfaz página web restaurante – Redes Sociales



Autor: Cadena, S. (2023)

Figura 101

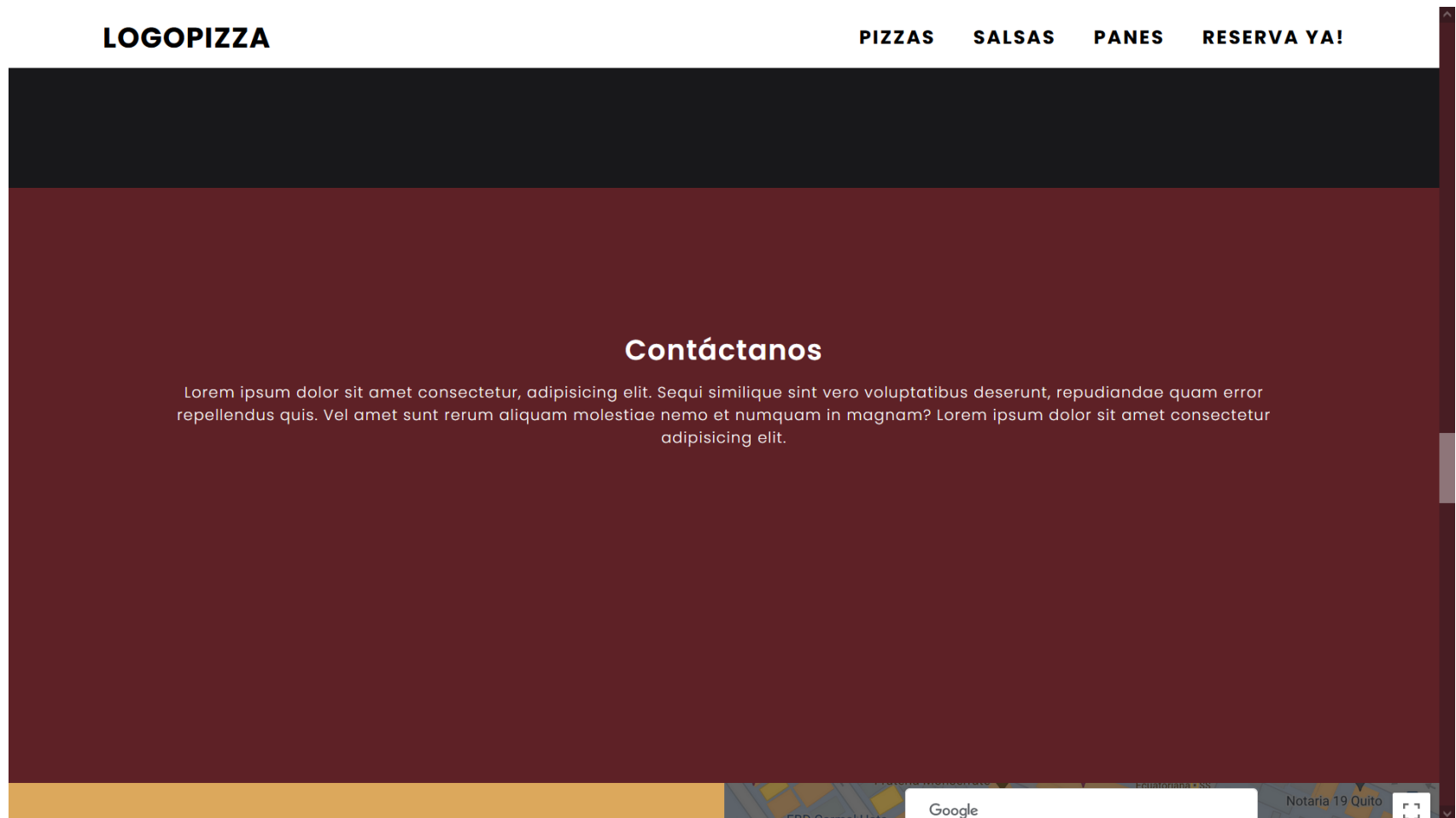
Interfaz página web restaurante - Comentarios



Autor: Cadena, S. (2023)

Figura 102

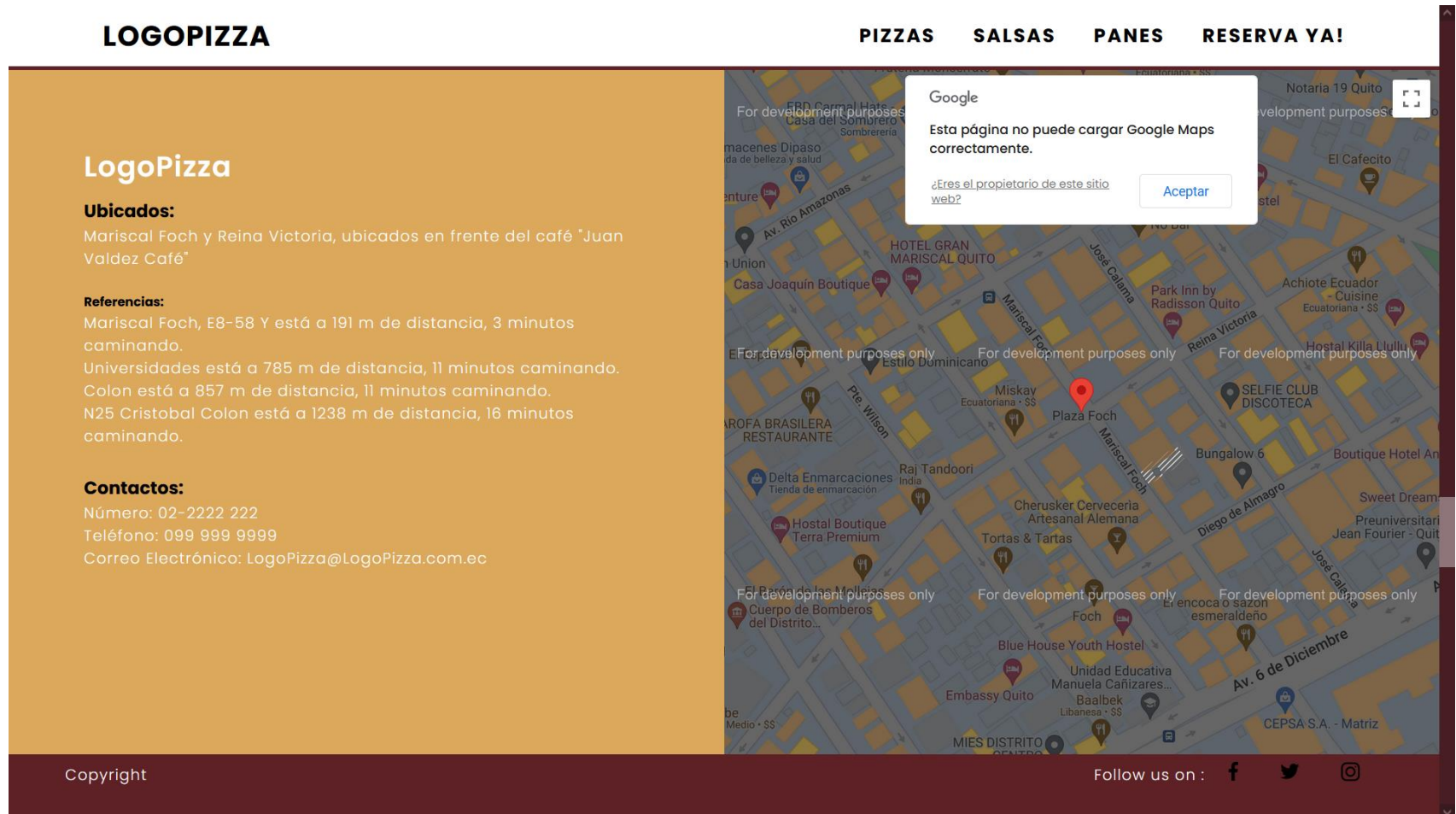
Interfaz página web restaurante - Contacto



Autor: Cadena, S. (2023)

Figura 103

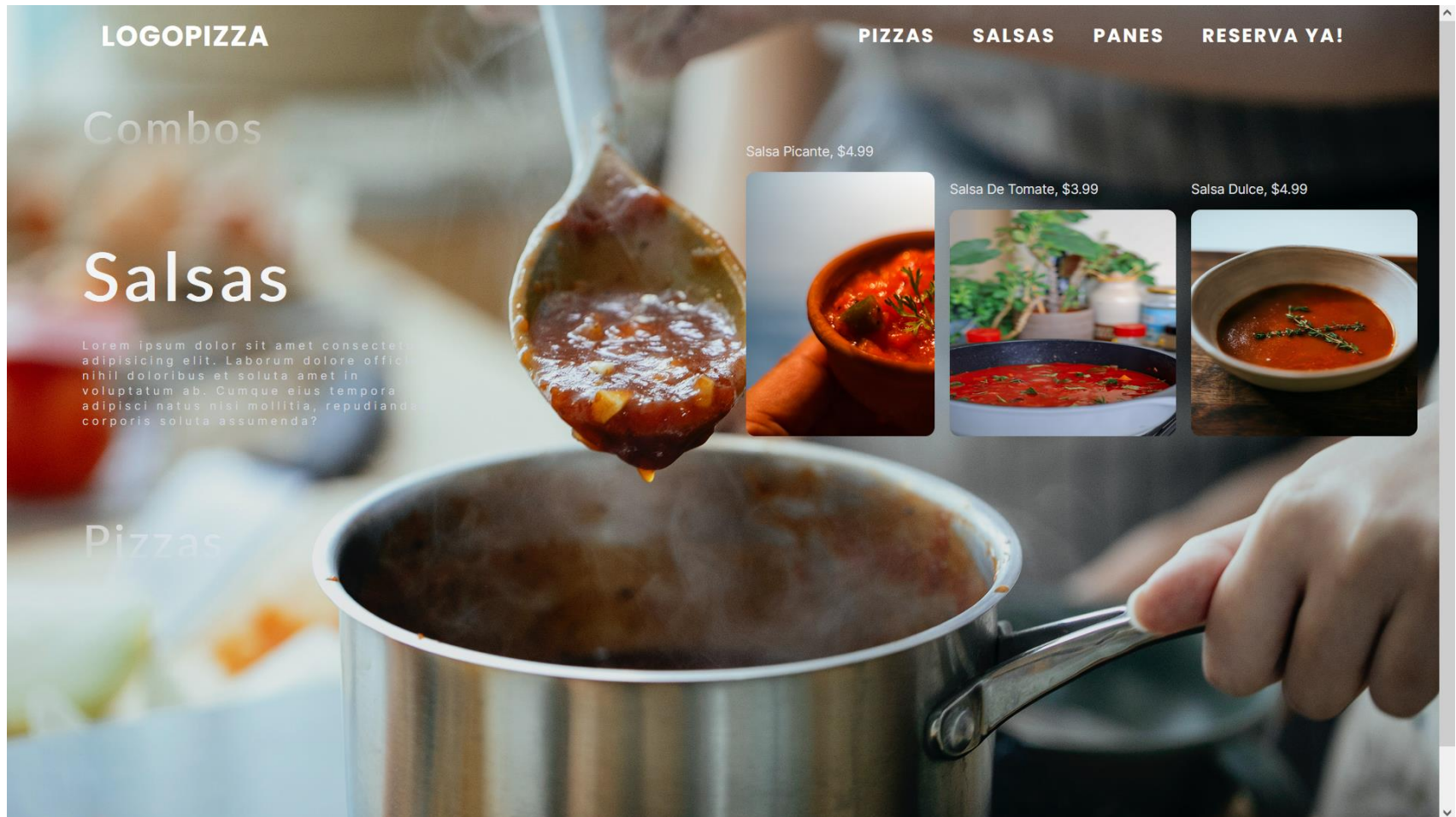
Interfaz página web restaurante - Ubicación



Autor: Cadena, S. (2023)

Figura 104

Interfaz página web restaurante - Platos salsas



Autor: Cadena, S. (2023)

Figura 105

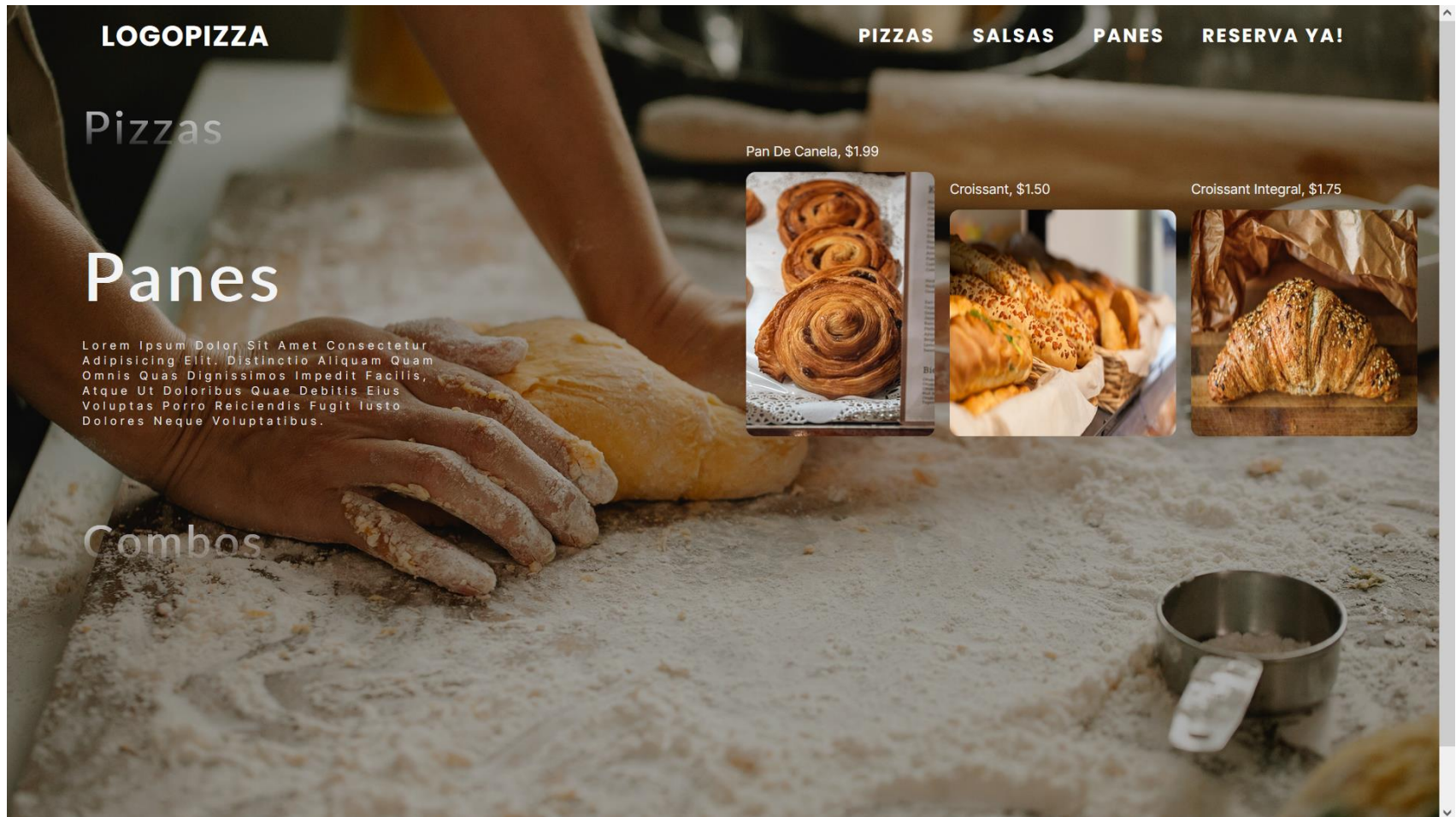
Interfaz página web restaurante - combos



Autor: Cadena, S. (2023)

Figura 106

Interfaz página web restaurante - Panes



Autor: Cadena, S. (2023)

Figura 107

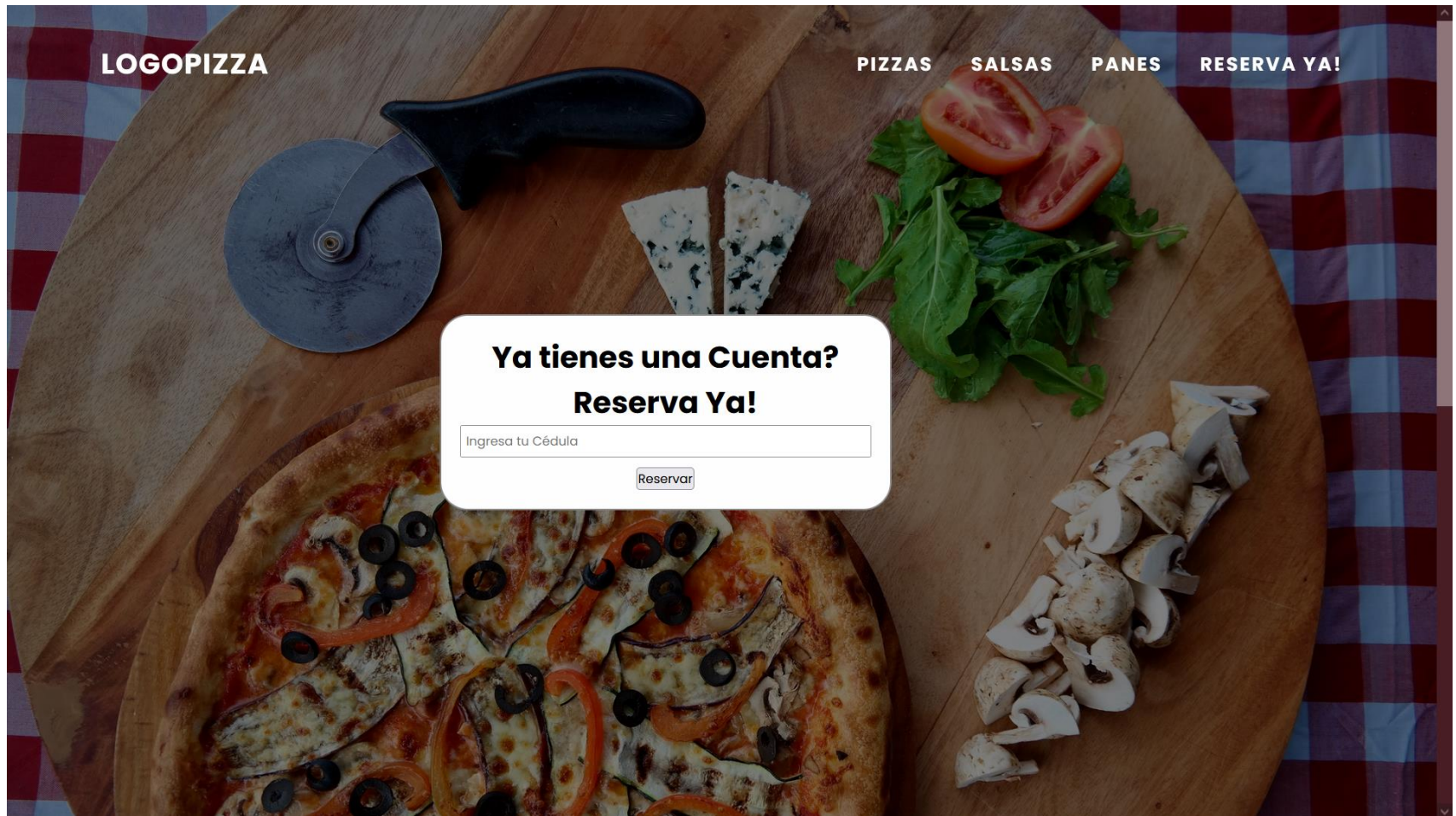
Interfaz página web restaurante - Pizzas



Autor: Cadena, S. (2023)

Figura 108

Interfaz página web restaurante - Reserva cliente



Autor: Cadena, S. (2023)

Figura 109

Interfaz página web restaurante - Registro reserva

The image shows a web interface for a restaurant named "LOGOPIZZA". The background features a wooden pizza board with a pizza, a pizza cutter, and fresh ingredients like tomatoes and mushrooms. The navigation menu includes "PIZZAS", "SALSAS", "PANES", and "RESERVA YA!". A white registration form titled "Regístrate" is overlaid on the page. The form contains the following fields and labels:

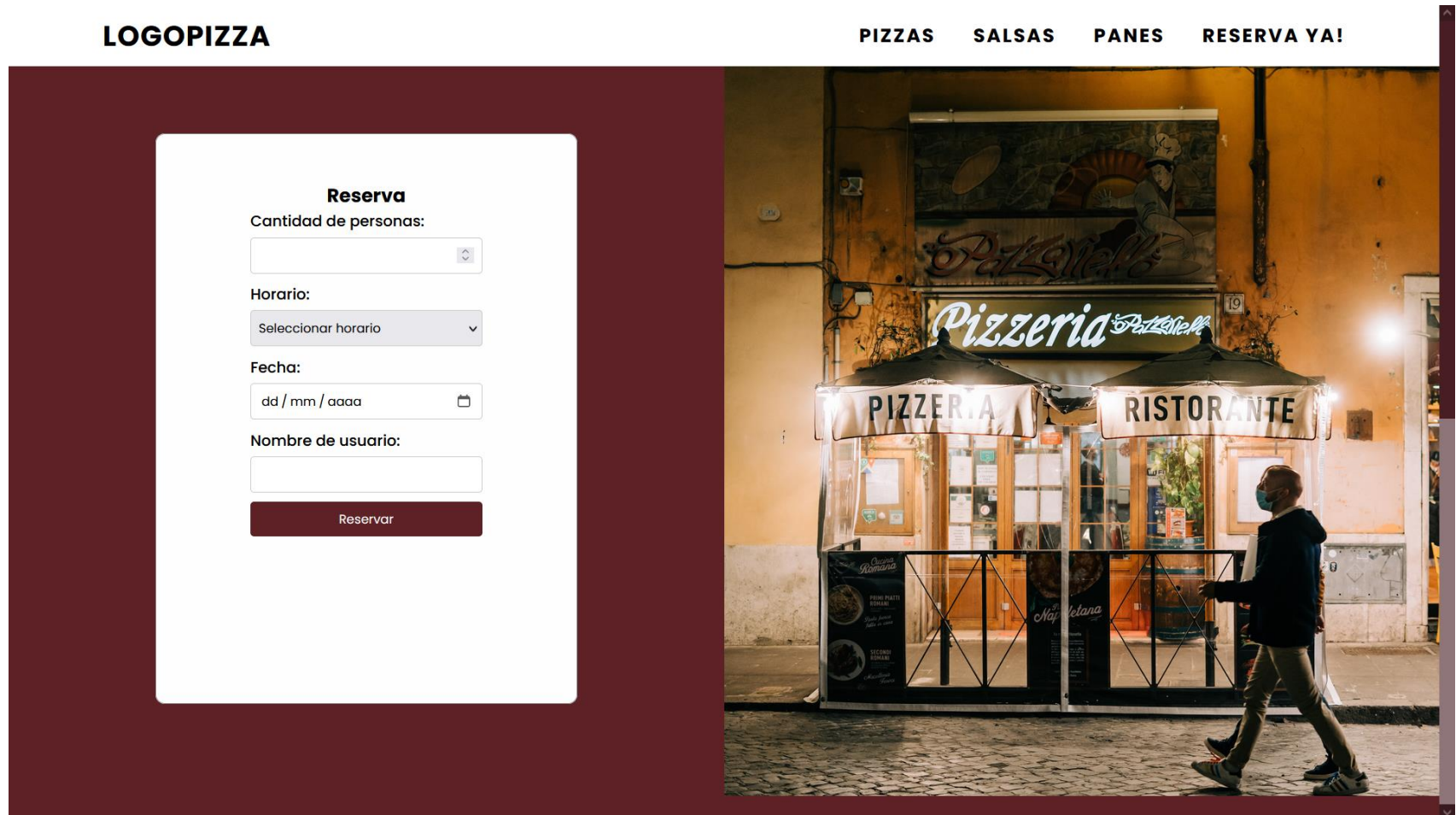
- Cedula: Ingresar tu Cédula
- Nombre: Ingresar tu Nombre
- Apellido: Ingresar tu Apellido
- Dirección: Ingresar tu Dirección
- Teléfono: Ingresar tu Teléfono
- Correo: Ingresar tu Correo

At the bottom of the form is a button labeled "Crear Cuenta".

Autor: Cadena, S. (2023)

Figura 110

Interfaz página web restaurante – Formulario Reserva



Autor: Cadena, S. (2023)